

Universitat de Lleida
Escola Politècnica Superior
Enginyeria Tècnica en informàtica de Gestió

Treball final de carrera

**Disseny i implementació del mòdul de generació de tests d'una aplicació
web d'autoavaluació desenvolupada en comunitat.**

Autor: Xavier Aiguabella i Guilera

Director: Josep Maria Ribó i Balust

Juliol de 2007

Agraïments

Agraeixo a totes aquelles persones que m'han donat suport per realitzar aquest projecte.

Primerament al meu tutor, Jospe Maria Ribó, a pesar de diversos problemes personals la seva ajuda a set inestimable, sense la seva ajuda no hagués set possible realitzar aquest projecte.

A la meva família, pares, germà, novia i amics, pels seus ànims i motivació constant.

A tots ells els hi estic molt agraït.

Índex

| | |
|---|----|
| 1. Introducció..... | 6 |
| 2. L'eina d'ajut a l'autoavaluació InnovaCampus..... | 9 |
| 2.1. Breu explicació | 9 |
| 2.2. Funcionalitat | 12 |
| 2.3. Tecnologies emprades..... | 17 |
| 3. Anàlisi de requeriments | 18 |
| 3.1. Requeriments generals i actors del sistema | 19 |
| 3.2. Casos d'ús | 22 |
| 3.3. Model de domini..... | 38 |
| 4. Aspectes de disseny i implementació..... | 41 |
| 4.1. Disseny de la Base de Dades | 42 |
| 4.2. Model Vista Controlador | 44 |
| 4.3. Classes de disseny | 46 |
| 4.3.1. Classes del Model | 46 |
| 4.3.1. Classes de la Vista | 54 |
| 4.3.1. Classes del Controlador | 58 |
| 4.4. Algorisme de dependència entre preguntes | 61 |
| 4.4.1. Idea bàsica | 61 |
| 4.4.2. Integració en l'aplicació | 65 |
| 4.5. Aspectes de la implementació | 69 |
| 5. Conclusió i treball futur | 71 |

| | |
|--|---------|
| Annex A: Tecnologies web | 74 |
| A.1. HTML..... | 74 |
| A.2. JavaScript..... | 77 |
| A.3. CSS | 79 |
| A.4. JSP..... | 82 |
| Annex B: Jakarta-Struts | 85 |
| Annex C: JGraphT..... | 88 |
| Annex D: Especificació..... | 90 |
| Annex E: Preguntes des de fitxer | 98 |
| Annex F: Llicència GPL..... | 102 |
| Llistat d'acrònims..... | 122 |
| Bibliografia..... | 123 |

Capítol 1

Introducció

Segons el Gran Diccionari de la llengua Catalana estudiar és aplicar l'esperit a adquirir la coneixença d'alguna cosa, examinar-la atentament per determinar-ne la natura, el caràcter, la significació, per interpretar-la, reproduir-la, etc.

L'estudiant ha d'estudiar una matèria i en un dia determinat avaluar el seu grau de coneixement adquirit. Una acció útil per l'estudiant es comprovar la seva coneixença sobre la matèria durant el seu període d'estudi. Així podria dedicar més hores a aquelles parts que considera que no té un bon nivell.

Una manera de comprovar aquest grau de coneixement es realitzant proves o test ell mateix de la matèria que s'ha de saber. Així podria saber en que s'equivoca i poder-ho estudiar més temps.

Hi ha un projecte anomenat InnovaCampus que permet a l'estudiant realitzar test sobre diverses assignatures universitàries i al professorat fer un seguiment del

grau d'assimilació de les matèries per part dels seus estudiants. En aquest projecte, el professor de l'assignatura crea unes quantes preguntes i l'estudiant pot realitzar test i comprovar en que s'ha equivocat.

El meu projecte consisteix en dissenyar i implementar una part del projecte InnovaCampus, implementant-la en noves tecnologies i augmentant la seva funcionalitat. Més concretament, consisteix en dissenyar i implementar un mòdul de generació de test de preguntes amb recursos i/o amb dependència entre preguntes.

Quan parlem de preguntes amb recursos, ens referim a que per poder resoldre la pregunta necessitem informació addicional, un recurs: una imatge, un vídeo, una reproducció en so (àudio), un document de text, etc. Quan es creï una pregunta es podrà triar l'opció de lligar-li un recurs; i quan aquesta pregunta formi part d'un test, es podrà visualitzar aquest recurs per poder resoldre correctament la pregunta.

Quan parlem de preguntes amb dependència ens referim a que una pregunta no pot formar part del test si abans no hi ha la pregunta de la qual en depèn. D'aquesta manera es pot generar un test amb un conjunt de preguntes que estan vinculades entre si.

Abans de començar a treball, o inclús pensar en el meu projecte, haure d'entendre la nova tecnologia que farem servir, que serà jakarta-struts. Un cop ja sàpiga utilitzar aquesta tecnologia, haure d'entendre el funcionament del projecte InnovaCampus. Finalment haure de dissenyar i implementar el meu mòdul fent servir aquesta tecnologia i el meu coneixement sobre InnovaCampus.

En aquest projecte, explicaré primer el projecte InnovaCampus, una eina d'ajut a l'autoavaluació. Seguidament faré un anàlisi dels requeriments necessaris per fer

el meu mòdul. Després parlaré sobre els aspectes de disseny i implementació. Per acabar una conclusió del projecte i alguns possibles projectes futurs relacionats amb el meu mòdul d'InnovaCampus i tot el projecte en si. A més a més, en els annexos es disposarà d'informació relacionada amb les tecnologies emprades per dur a terme el meu projecte.

Capítol 2

L'eina d'ajut a l'autoavaluació InnovaCampus

2.1. Breu explicació

InnovaCampus és un portal web que permet fer test d'autoavaluació i genera unes estadístiques relacionades amb els tests realitzats.

Aquests portal web és un software tant per a professors com per a alumnes. Els alumnes poden realitzar tests sobre una matèria en concret, i els professors obtenen unes estadístiques per poder saber el grau de coneixement de l'alumnat sobre la matèria en qüestió.

L'aplicació té tres tipus d'usuaris: l'administrador, els professors i els alumnes. L'administrador serà l'encarregat de gestionar tot el que un professor pugui fer.

Aquesta aplicació ajuda tant al professor com a l'alumne. Al professor li permet fer un seguiment dels alumnes respecte l'assignatura i els seus progressos; mentre que per l'alumne serà un mecanisme d'auto-aprenentatge, ja que podrà realitzar test sempre que ho desitgi i al obtenir les respostes errònies podrà revisar i dedicar més temps a aquests aspectes que encara no ha assolit.

El software esta dividit en professors i alumnes que seran diferenciats i identificats per realitzar les funcions especificques de cadascú. Les assignatures, que un professor imparteix o un alumnes està matriculat, estan dividides en temes i, a la vegada, cada tema esta dividit en objectius.

El professor és l'encarregat de crear les noves assignatures, amb els temes i els objectius necessaris, i de donar d'alta als alumnes a les assignatures que estan matriculats, donant accés als alumnes a la realització dels test dels objectius corresponents a l'assignatura.

Els tests es generen seleccionant aleatòriament preguntes d'un determinat objectiu. Per poder fer uns tests complets, el professor han de crear diverses preguntes per cada objectiu, fent que d'aquesta manera els alumnes puguin realitzar tests diferents cada vegada. Una pregunta esta formada per un enunciat, i un nombre determinat de respostes possibles que el professor pot triar quan crea o edita la pregunta.

Per a que un alumne pugui realitzar un test, haurà de triar l'objectiu d'un tema determinat dins una assignatura. L'alumne anirà visualitzant les preguntes i triant les respostes que cregui correctes. Cada resposta correcta sumarà uns determinats punts, les respostes incorrectes restaran uns determinats punts, i si el alumnes decideix no respondre la pregunta, no es sumarà o restarà cap punt. Al finalitzar el test l'alumne podrà veure la seva puntuació i les respostes correctes i incorrectes. Quan visualitzi les respostes, es podrà veure els comentaris que el

professor a ficat d'acord a les respostes seleccionades, i l'alumne s'adonará de quines parts del temari desconeix o no seu sap del tot bé.

Un cop es finalitza un test, es generen o s'actualitzen unes estadístiques que el professor pot veure. Aquestes estadístiques són molt importants per al professor, ja que li informen sobre la assignatura i també sobre un alumne en concret. Aquesta funció permet saber quines parts de la matèria estan assolides pels estudiants i quines parts no. També permet saber el grau de coneixement i l'evolució d'un alumne en concret. Les estadístiques permeten al professor portar un control del grau de coneixement de la matèria en els estudiants.

Com s'ha pogut veure, la funcionalitat d'InnovaCampus és intuïtiva i fàcil d'utilitzar i a més a més és una eina molt útil pel professor i l'alumne, i ajuda tant a un com a l'altre.

2.2. Funcionalitat

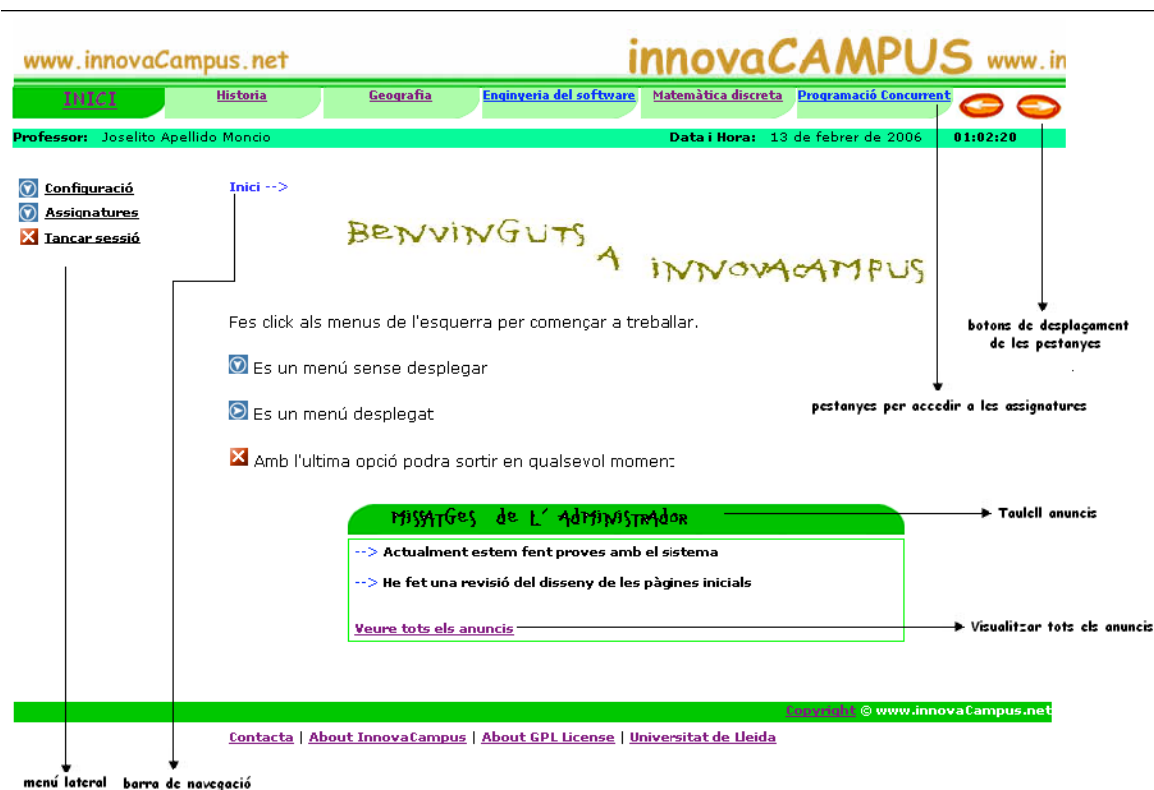
Funcions de l'administrador en InnovaCampus:



Captura de la pantalla inicial de l'administrador.

- Donar d'alta/baixa als professors.
- Canviar les seves dades: Login i Password.
- Donar d'alta/baixa anuncis. Els anuncis podran anar dirigits a alumnes, professors, a ambdós, o podran ser anuncis generals per a usuaris i per aquells que no accedeixin com a usuaris

Funcions dels professors en InnovaCampus:



Captura de la pantalla inicial dels professors.

- Canviar les seves dades: Nom, Cognom, Email, Telèfon, Login i Password.
- Pot gestionar assignatures.
- Pot gestionar professors.
- Pot gestionar temes.
- Pot gestionar objectius.

The screenshot shows the InnovaCampus web interface. At the top, there's a header with the website name and a navigation bar with tabs for different subjects: INICI, Historia, Geografia, Enginyeria del software (selected), Matemàtica discreta, and Programació Concurrent. Below the header, it displays the professor's name (Joselito Apellido Mondio) and the date/time (13 de febrer de 2006, 01:11:43).

On the left side, there's a sidebar menu with options: Professors, Temes, Objectius, Preguntes, Respostes, Alumnus, Anuncis, Estadístiques, and Tancar sessió. A vertical line points from the 'menú lateral' label to this sidebar.

The main content area is titled 'Enginyeria del software'. It includes a sub-header 'Inici --> Assignatura' and a message: 'Fes click als menus de l'esquerra per començar a treballar amb l' assignatura seleccionada.' Below this, there are three radio buttons:

- ☐ Es un menú sense desplegar
- ☐ Es un menú desplegat
- ☒ Amb l'ultima opció podra sortir en qualsevol moment

In the center, there's a green box titled 'Tauler d'anuncis de l'assignatura'. Inside, it says '--> s'han introduït nous objectius en el Tema Patrons de Disseny'. Below this, there's a link 'Veure tots els anuncis'. To the right of the box, there are two arrows: one pointing to 'Tauler d'anuncis' and another pointing to 'Visualitzar tots els anuncis'.

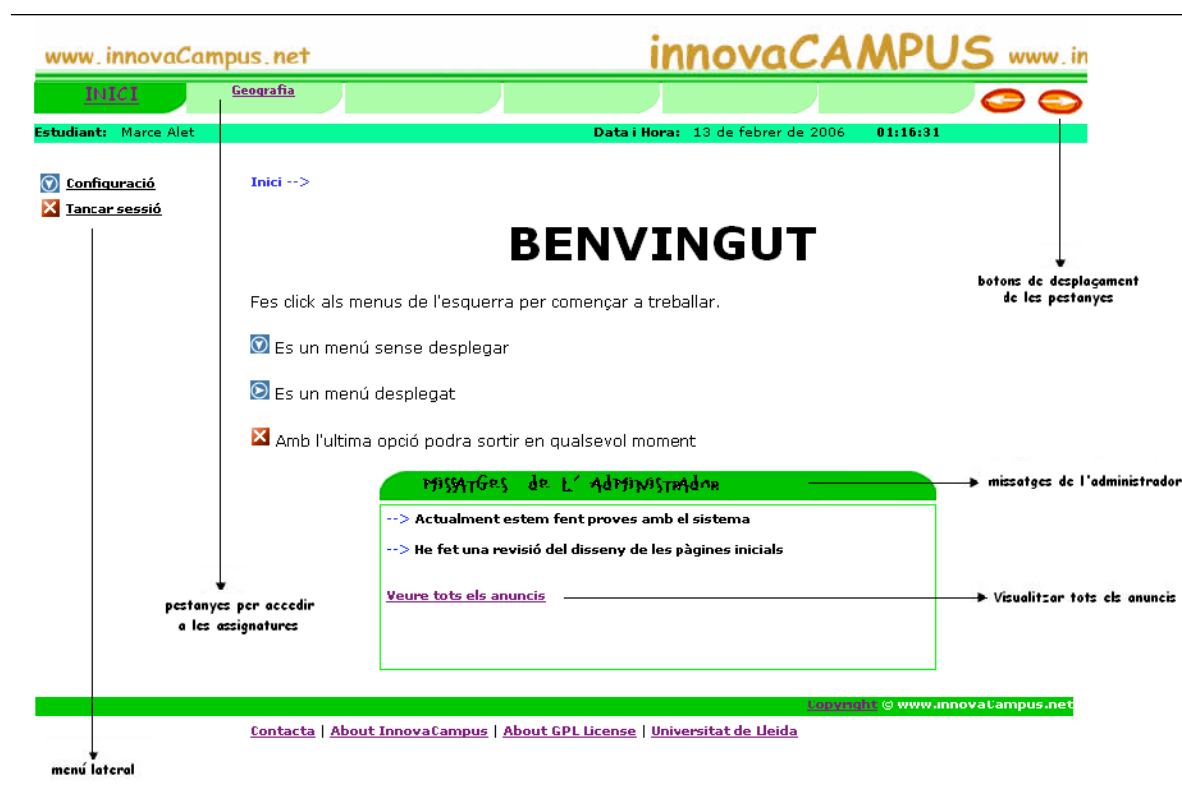
At the bottom, there's a footer with links: Contacta, About InnovaCampus, About GPL License, and Universitat de Lleida. A copyright notice 'Copyright © www.innovaCampus.net' is also present.

Camp de treball d'una assignatura d'un professor.

- Pot gestionar preguntes.
- Pot gestionar respostes.
- Permet reinicialitzar una assignatura.
- Permet matricular alumnes, ja sigui fent l'alta manualment des de teclat o mitjançant un fitxer amb un format determinat.
- Pot visualitzar estadístiques que fan referència als diferents tests realitzats pels estudiants d'una assignatura. Les estadístiques podran ser referents a un estudiant concret, o més globals, referents a un objectiu concret d'un tema de l'assignatura.

- Pot gestionar el taulell d'anuncis de l'assignatura.
- Pot visualitzar anuncis de l'administrador.

Funcions dels estudiants en InnovaCampus:



Pantalla Inicial dels alumnes

- Visualitzar anuncis de l'administrador.
- Visualitzar el taulell d'anuncis de les assignatures.

www.innovaCampus.net

innovaCAMPUS www.in

INICI

Professor: Marce Alet

Data i Hora: 13 de febrer de 2006 01:21:42

[Inici --> Test --> Llista Temes --> Llista Objectius --> Resoldre Test](#)

☒ Testos

☒ Anuncis

☒ Tancar sessió

RESOL EL TEST

ELEGEIX LA RESPOSTA CORRECTA PER A CADA QUESTIO.

Temps restant per resoldre el test: 9:46

Temps per resoldre el test

accés a la pregunta anterior

ANTERIOR

QUESTIO 1/2

SEGÜENT

accés a la següent pregunta

text de la pregunta

1- Mejor grupo de la historia?

respostes de la pregunta

☐ Led Zeppelin

☐ Deep purple

☐ Ages of empires

☐ Guns and roses

número de preguntes

FINALITZAR TEST

per finalitzar el test

Contacta | About InnovaCampus | About GPL License | Universitat de Lleida

Copyright © www.innovaCampus.net

Pantalla de resolució d'un test per part d'un estudiant.

- Realitzar tests.
- Visualitzar els resultats d'un test.

2.3. Tecnologies emprades

La tecnologia més rellevant sota la qual s'ha desenvolupat el projecte InnovaCampus és la tecnologia JSP, una tecnologia basada en Java que simplifica el desenvolupament d'aplicacions web amb contingut dinàmic.

Les tecnologies utilitzades en el desenvolupament i dissenyen de les pàgines web són el llenguatge de marques HTML i el llenguatge interpretat JavaScript. El llenguatge JavaScript ha estat important per a desenvolupar aspectes molt destacats de cara a la capa de presentació de l'aplicació.

Capítol 3

Anàlisi de requeriments

En aquest capítol farem un anàlisi dels requeriments necessaris per dissenyar i implementar un mòdul de generació de test de preguntes amb recursos i/o amb dependència entre preguntes. Consisteix en establir els serveis que el sistema haurà de proporcionar a l'usuari, es a dir, s'ha de determinar que es allò que el sistema pot fer i com ho ha de fer.

Per fer això, primer establim a grans trets quins son els requeriments que tenim, després analitzarem aquests requeriments i els escriurem en forma de casos d'ús. Un cop tenim tots els casos d'ús necessaris, crearem el model de domini que presentarà el nostre mòdul.

3.1. Requeriments generals i actors del sistema

Hem de dissenyar un mòdul de generació de test de preguntes amb recursos i/o amb dependència entre preguntes.

Els tests que un alumne pot realitzar estan compostats d'un nombre determinat de preguntes. Amb aquest nou mòdul, hi haurà preguntes que aniran acompanyades de determinats recursos, podrà ser arxius de reproducció de so, vídeos, imatges o altres arxius com documents de text. L'alumne veurà la pregunta amb les respostes, i podrà accedir a visualitzar el recurs associat a la pregunta. Amb l'ajuda de la informació proporcionada pel recurs, l'alumne podrà resoldre la pregunta més fàcilment.

La idea de la dependència entre preguntes ens permet tenir la possibilitat de que es generin tests més interessants. Ja no parlem d'uns tests amb un conjunt de preguntes i ja està; ara parlem d'uns tests amb preguntes relacionades entre si. La dependència entre preguntes ens dona la possibilitat de crear preguntes que per poder respondre-les correctament necessites que abans hagi sortit una altra pregunta, ja sigui pel resultat, els valors, la manera de pensar la solució, per comparar preguntes o resultats...

Com hem vist, tenim casos, per una banda que les preguntes poden estar o no associades a un recurs, i per una altra banda que les preguntes poden dependre d'unes altres preguntes.

El primer cas ens diu que es pot associar un recurs a una pregunta. Això ens planteja les següents idees:

- Les preguntes puguin estar relacionades amb els recursos.
- Incorporar i gestionar el objecte recurs dins del sistema d'InnovaCampus.
- Visualitzar un recurs determinat.

En el segon cas tenim que les preguntes poden dependre d'altres preguntes. Les idees que ens planteja aquest segon cas són:

- Incorporar i gestionar la idea de dependència entre preguntes dins del sistema InnovaCampus.
- Establir una relació de dependència entre una pregunta i una altra.
- Generar un test d'acord amb les dependències existents entre les preguntes de l'objectiu determinat.

D'acord al projecte InnovaCampus, el sistema esta format per tres actors: administrador, professor i alumne. En el nostre cas, el actor administrador no es necessari ja que no el fem servir i per tant ens quedem solament amb els actors professor i alumne, que seran els usuaris del nostre mòdul.

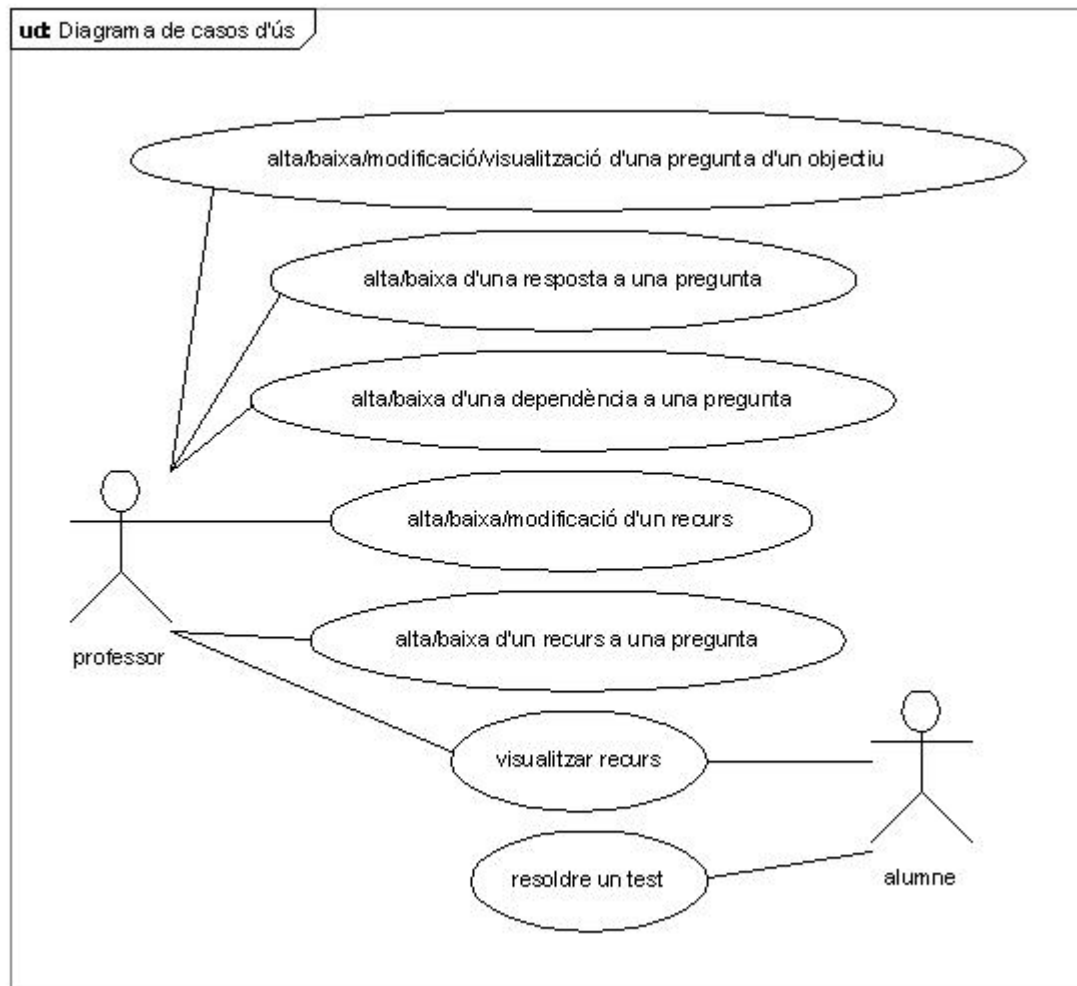
Funcions del professor:

- Gestionar preguntes.
- Gestionar respostes.
- Gestionar recursos.
- Visualitzar recursos.
- Gestionar la dependència entre preguntes.

Funcions de l'alumne:

- Realitzar tests.
- Visualitzar recursos.

A partir d'un diagrama de casos d'ús podem observar de manera més gràfica els usuaris i les funcions que exerceixen.



3.2. Casos d'ús

Els casos d'ús descriuen el comportament d'un sistema de del punt de vista de l'usuari. Com hem vist en l'apartat anterior, el nostre sistema estava format per dos actors, el professor i el alumne. Primer llistem els casos d'ús de cada actor i després detallem cada cas d'us per separat.

Casos d'ús del professor:

- Donar d'alta una pregunta a un objectiu.
- Donar de baixa una pregunta d'un objectiu.
- Modificar una pregunta d'un objectiu.
- Visualitzar una pregunta.
- Donar d'alta una resposta d'una pregunta.
- Donar de baixa una resposta d'una pregunta.
- Donar d'alta una dependència a una pregunta.
- Donar de baixa una dependència a una pregunta.
- Donar d'alta un recurs a un objectiu.
- Donar de baixar un recurs d'un objectiu.
- Modificar un recurs d'un objectiu.
- Donar d'alta un recurs a una pregunta.
- Donar de baixa un recurs a una pregunta.
- Visualitzar un recurs.

Casos d'ús de l'alumne:

- Resoldre un test.
- Visualitzar un recurs.

| | | |
|-------------------------|---|---|
| Nom | Donar d'alta una pregunta a un objectiu | |
| Descripció | El professor dona d'alta una pregunta dins d'un objectiu en un dels temes d'una assignatura | |
| Actors | Professor | |
| Precondicions | El professor es troba dins l'aplicació i vol afegir una pregunta a un objectiu | |
| Postcondicions | La pregunta s'ha afegit a l'objectiu determinat | |
| Seqüència normal | Professor | Sistema |
| | 1. Accedeix a l'espai de treball de l'assignatura. | 2. Registra la situació. |
| | 3. Indica que vol afegir una pregunta. | 4. Registra la situació i genera una llista dels temes de l'assignatura. |
| | 5. Selecciona el tema. | 6. Registra la situació i genera una llista dels objectius del tema. |
| | 7. Selecciona l'objectiu. | 8. Registra la situació i demana les dades necessàries per donar d'alta una pregunta. |
| | 9. Introdueix totes les dades, el text de la pregunta, les respostes, si vol associar un recurs, establir dependències. | 10. Es dona d'alta la pregunta. |
| | Casos d'ús a incloure: | |
| | Donar d'alta una resposta d'una pregunta. | |
| | Donar d'alta un recurs a una pregunta. | |
| | Donar d'alta una dependència a una pregunta. | |
| | 11. Visualitza la nova pregunta creada. | |
| Excepcions | 10a. Falten dades per poder donar d'alta la pregunta. 10a1. El professor acaba d'introduir totes les dades. | |

| | | |
|-------------------------|--|--|
| Nom | Donar de baixa una pregunta d'un objectiu | |
| Descripció | El professor dona de baixa una pregunta determinada. També s'eliminaran les respostes, les dependències, els resultats i els tests associats | |
| Actors | Professor | |
| Precondicions | El professor es troba dins l'aplicació i vol eliminar una pregunta d'un objectiu | |
| Postcondicions | S'ha eliminat la pregunta i les respostes, dependències, resultats i tests relacionats | |
| Seqüència normal | Professor | Sistema |
| | 1. Accedeix a l'espai de treball de l'assignatura. | 2. Registra la situació. |
| | 3. Indica que vol eliminar una pregunta. | 4. Registra la situació i genera una llista dels temes de l'assignatura. |
| | 5. Selecciona el tema. | 6. Registra la situació i genera una llista dels objectius del tema. |
| | 7. Selecciona l'objectiu. | 8. Registra la situació i genera una llista dels preguntes de l'objectiu |
| | 9. Selecciona la pregunta | 10. Se li demana confirmació. |
| | 11. Accepta la confirmació. | 12. S'elimina la pregunta. |
| | 13. Visualitza una llista amb la resta de preguntes de l'objectiu | |

| | | |
|-------------------------|--|--|
| Nom | Modificar una pregunta d'un objectiu | |
| Descripció | El professor modifica les dades d'una pregunta determinada | |
| Actors | Professor | |
| Precondicions | El professor es troba dins l'aplicació i vol modificar les dades d'una pregunta | |
| Postcondicions | S'han modificat les dades correctament | |
| Seqüència normal | Professor | Sistema |
| | 1. Accedeix a l'espai de treball de l'assignatura. | 2. Registra la situació. |
| | 3. Indica que vol modificar una pregunta. | 4. Registra la situació i genera una llista dels temes de l'assignatura. |
| | 5. Selecciona el tema. | 6. Registra la situació i genera una llista dels objectius del tema. |
| | 7. Selecciona l'objectiu. | 8. Registra la situació i genera una llista dels preguntes de l'objectiu |
| | 9. Selecciona la pregunta | 10. Obté totes les dades relacionades amb la pregunta. |
| | 11. Modifica el text de la pregunta, les respostes, el recurs associat, les dependències. | 12. Es guarden els canvis. |
| | Casos d'ús a incloure: | |
| | Donar d'alta/baixa una resposta d'una pregunta. | |
| | Donar d'alta/baixa un recurs a una pregunta. | |
| | Donar d'alta/baixa una dependència a una pregunta. | |
| | 12. Visualitza els canvis modificats. | |
| Excepcions | 12a. Falten dades de la pregunta. 12a1. El professor acaba d'introduir totes les dades. | |

| | | |
|-------------------------|---|---|
| Nom | Visualitzar una pregunta | |
| Descripció | El professor visualitza les dades d'una pregunta determinada | |
| Actors | Professor | |
| Precondicions | Professor: es troba dins l'aplicació i vol visualitzar una pregunta que ja està donada d'alta | |
| Postcondicions | Ha visualitzat la pregunta | |
| Seqüència normal | Professor | Sistema |
| | 1. Accedeix a l'espai de treball de l'assignatura. | 2. Registra la situació. |
| | 3. Indica que vol visualitzar una pregunta. | 4. Registra la situació i genera una llista dels temes de l'assignatura. |
| | 5. Selecciona el tema. | 6. Registra la situació i genera una llista dels objectius del tema. |
| | 7. Selecciona l'objectiu. | 8. Registra la situació i genera una llista de les preguntes de l'objectiu. |
| | 9. Selecciona la pregunta. | 10. Obté les dades. |
| | 11. Visualitza la pregunta, amb el text de la pregunta, les respostes, el recurs associat i les dependències. | |

| | | |
|-------------------------|---|--|
| Nom | Donar d'alta una resposta d'una pregunta | |
| Descripció | El professor afegeix una resposta a una pregunta determinada | |
| Actors | Professor | |
| Precondicions | El professor es troba en la edició d'una pregunta, tant pot ser en el moment de crear-la com en el de modificar-la | |
| Postcondicions | La resposta s'ha inclòs a la pregunta | |
| Seqüència normal | Professor | Sistema |
| | 1. Introdueix les dades per crear la nova resposta, el text de la resposta, el comentari i dir si és una resposta correcta o no | 2. Es processen les dades. |
| | 3. Acaba de crear/modificar la pregunta. | 4. Es guarda la pregunta amb la nova resposta. |
| | 5. Visualitza la pregunta amb la resposta creada. | |
| Excepcions | 4a. Falten dades de la resposta. 10a1. El professor acaba d'introduir totes les dades. | |

| | | |
|-------------------------|--|---|
| Nom | Donar de baixa una resposta d'una pregunta | |
| Descripció | El professor elimina una resposta d'una pregunta determinada | |
| Actors | Professor | |
| Precondicions | El professor es troba en la modificació de la pregunta | |
| Postcondicions | La resposta s'ha eliminat | |
| Seqüència normal | Professor | Sistema |
| | 1. Borra totes les dades de la resposta, el text, el comentari i ficar-la com no correcta. | 2. Es treu la resposta. |
| | 3. Acaba de modificar la pregunta. | 4. Es guarda la pregunta eliminant la resposta. |
| | 5. Visualitza la pregunta sense la resposta. | |

| | | |
|-------------------------|--|--|
| Nom | Donar d'alta una dependència a una pregunta | |
| Descripció | El professor afegeix una dependència a una pregunta determinada | |
| Actors | Professor | |
| Precondicions | El professor es troba en la edició d'una pregunta, tant pot ser en el moment de crear-la com en el de modificar-la | |
| Postcondicions | La resposta s'ha inclòs a la pregunta | |
| Seqüència normal | Professor | Sistema |
| | 1. Selecciona que vol ficar dependència. | 2. Es genera una llista amb les possibles preguntes. |
| | 3. Selecciona la preguntes amb la que vol establir la dependència. | 4. Es processen les dades. |
| | 5. Acaba de crear/modificar la pregunta. | 6. Es guarda la pregunta amb la nova dependència. |
| | 7. Visualitza la pregunta amb la dependència. | |
| Excepcions | 1a. Ja hi ha alguna altra dependència. 1a1. Passem directament al punt 3. 2a. No hi ha cap pregunta possible per establir una dependència. 2a1. No es pot donar d'alta una dependència. | |

| | | |
|-------------------------|---|--|
| Nom | Donar de baixa una dependència a una pregunta | |
| Descripció | El professor elimina una dependència a una pregunta determinada | |
| Actors | Professor | |
| Precondicions | El professor es troba en la modificació de la pregunta | |
| Postcondicions | La dependència s'ha eliminat de la pregunta | |
| Seqüència normal | Professor | Sistema |
| | 1. Deselecciona la dependència a eliminar. | 2. Es processen les dades. |
| | 3. Acaba de modificar la pregunta. | 4. Es guarda la pregunta eliminant la dependència. |
| | 5. Visualitza la pregunta sense la dependència. | |

| | | |
|-------------------------|--|--|
| Nom | Donar d'alta un recurs a un objectiu | |
| Descripció | El professor dona d'alta un recurs dins d'un objectiu en un dels temes d'una assignatura | |
| Actors | Professor | |
| Precondicions | El professor es troba dins l'aplicació i vol afegir un recurs a un objectiu | |
| Postcondicions | El recurs s'ha afegit a l'objectiu determinat | |
| Seqüència normal | Professor | Sistema |
| | 1. Accedeix a l'espai de treball de l'assignatura. | 2. Registra la situació. |
| | 3. Indica que vol afegir un recurs. | 4. Registra la situació i genera una llista dels temes de l'assignatura. |
| | 5. Selecciona el tema. | 6. Registra la situació i genera una llista dels objectius del tema. |
| | 7. Selecciona l'objectiu. | 8. Registra la situació i demana les dades necessàries per donar d'alta un recurs. |
| | 9. Introdueix el tipus de recurs, el nom i el fitxer. | 10. Es dona d'alta el recurs. |
| Excepcions | 10a. Falten dades, o estan incorrectes, per poder donar d'alta el recurs. | |
| | 10a1. El professor acaba d'introduir totes les dades. | |

| | | |
|-------------------------|---|--|
| Nom | Donar de baixa un recurs d'un objectiu | |
| Descripció | El professor dona de baixa un recurs determinat. També s'eliminaran les preguntes relacionades amb aquest recurs (Cas d'ús: Donar de baixa una pregunta d'un objectiu.) | |
| Actors | Professor | |
| Precondicions | El professor es troba dins l'aplicació i vol eliminar un recurs d'un objectiu | |
| Postcondicions | S'ha eliminat el recurs i les preguntes relacionades | |
| Seqüència normal | Professor | Sistema |
| | 1. Accedeix a l'espai de treball de l'assignatura. | 2. Registra la situació. |
| | 3. Indica que vol eliminar un recurs.. | 4. Registra la situació i genera una llista dels temes de l'assignatura. |
| | 5. Selecciona el tema. | 6. Registra la situació i genera una llista dels objectius del tema. |
| | 7. Selecciona l'objectiu. | 8. Registra la situació i genera una llista dels recursos de l'objectiu. |
| | 9. Selecciona el recurs | 10. Se li demana confirmació. |
| | 11. Accepta la confirmació. | 12. S'elimina el recurs. |
| | 13. Visualitza una llista amb la resta de recursos de l'objectiu | |

| | | |
|-------------------------|--|--|
| Nom | Modificar un recurs d'un objectiu | |
| Descripció | El professor modifica el nom d'un recurs determinat | |
| Actors | Professor | |
| Precondicions | El professor es troba dins l'aplicació i vol modificar el nom d'un recurs determinat | |
| Postcondicions | S'ha modificat el nom correctament | |
| Seqüència normal | Professor | Sistema |
| | 1. Accedeix a l'espai de treball de l'assignatura. | 2. Registra la situació. |
| | 3. Indica que vol modificar un recurs. | 4. Registra la situació i genera una llista dels temes de l'assignatura. |
| | 5. Selecciona el tema. | 6. Registra la situació i genera una llista dels objectius del tema. |
| | 7. Selecciona l'objectiu. | 8. Registra la situació i genera una llista dels recursos de l'objectiu |
| | 9. Selecciona el recurs. | 10. Obté el nom actual relacionat amb el recurs. |
| | 11. Modifica el nom i/o el fitxer font. | 12. Es guarda el canvi. |
| | 12. Visualitza el canvi modificat. | |
| Excepcions | 12a. El nom ja existeix o no s'ha introduït. 12a1. Tornem al pas 11. | |

| | | |
|-------------------------|---|---|
| Nom | Donar d'alta un recurs a una pregunta | |
| Descripció | El professor vol relacionar un recurs a una pregunta | |
| Actors | Professor | |
| Precondicions | El professor es troba en la edició d'una pregunta, tant pot ser en el moment de crear-la com en el de modificar-la. El recurs ja s'ha donat d'alta | |
| Postcondicions | El recurs ja no està relacionat amb la pregunta | |
| Seqüència normal | Professor | Sistema |
| | 1. Selecciona que vol relacionar un recurs. | 2. Es genera una llista amb els tipus de recursos. |
| | 3. Selecciona el tipus de recurs que vol. | 4. Es genera una llista amb els possibles recursos. |
| | 5. Selecciona el recurs | 4. Es processen les dades. |
| | 5. Acaba de crear/modificar la pregunta. | 6. Es guarda la pregunta amb la relació al recurs. |
| | 7. Visualitza la pregunta amb el recurs seleccionat. | |
| Excepcions | 1a. Ja hi ha algun altre recurs relacionat. 1a1. Seleccionem el nou tipus i/o recurs que volem. 2a. No hi ha cap recurs donat d'alta. 2a1. No es pot relacionar el recurs. | |

| | | |
|-------------------------|---|--|
| Nom | Donar de baixa un recurs a una pregunta | |
| Descripció | El professor vol eliminar la relació d'un recurs a una pregunta | |
| Actors | Professor | |
| Precondicions | El professor es troba en la modificació de la pregunta | |
| Postcondicions | La relació al recurs s'ha eliminat de la pregunta | |
| Seqüència normal | Professor | Sistema |
| | 1. Deselecciona la opció d'associar un recurs. | 2. Es processen les dades. |
| | 3. Acaba de modificar la pregunta. | 4. Es guarda la pregunta eliminant la relació al recurs. |
| | 5. Visualitza la pregunta sense el recurs. | |

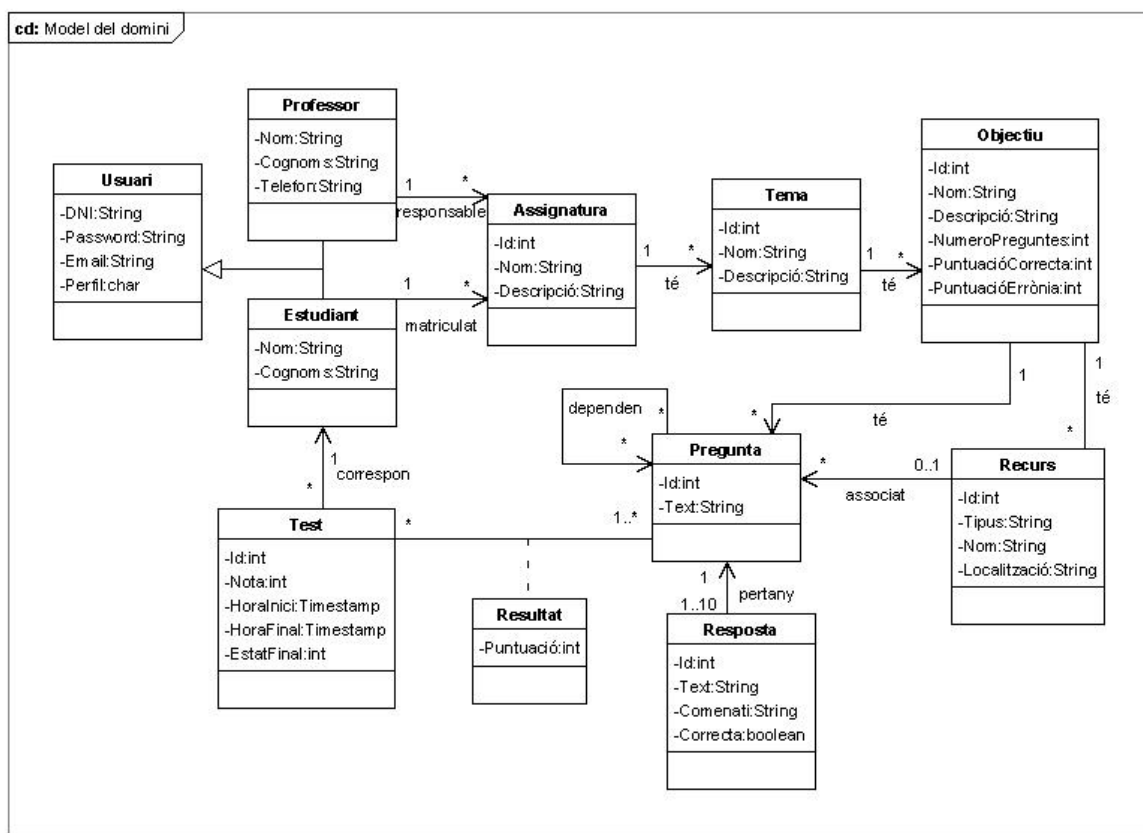
| | | |
|-------------------------|---|--|
| Nom | Visualitzar un recurs | |
| Descripció | Es visualitza un recurs determinat, ja sigui imatge, àudio, vídeo o un document | |
| Actors | Professor, alumne | |
| Precondicions | Professor: es troba dins l'aplicació i vol visualitzar un recurs determinat Alumne: esta realitzant un test i té l'opció de visualitzar un recurs. | |
| Postcondicions | El recurs s'ha pogut visualitzar | |
| Seqüència normal | Professor | Sistema |
| | 1. Accedeix a l'espai de treball de l'assignatura. | 2. Registra la situació. |
| | 3. Indica que vol visualitzar un recurs. | 4. Registra la situació i genera una llista dels temes de l'assignatura. |
| | 5. Selecciona el tema. | 6. Registra la situació i genera una llista dels objectius del tema. |
| | 7. Selecciona l'objectiu. | 8. Registra la situació i genera una llista dels recursos de l'objectiu. |
| | 9. Selecciona el recurs. | 10. Obté el recurs. |
| | 11. Visualitza el recurs. | |
| | Alumne | Sistema |
| | 1. Selecciona l'opció de visualitzar un recurs. | 2. Obté el recurs. |
| | 3. Visualitza el recurs. | |

| | | |
|-------------------------|---|--|
| Nom | Resoldre un test | |
| Descripció | L'alumne té un determinat temps per resoldre totes les preguntes del test. Va escollint les respostes que creu correctes de les diferents preguntes fins a completar-lo. Al final se li mostra un informe sobre el test amb la puntuació i les respostes escollides | |
| Actors | Alumne | |
| Precondicions | L'alumne vol realitzar un test | |
| Postcondicions | El test finalitza | |
| Seqüència normal | Alumne | Sistema |
| | 1. Accedeix a l'espai de treball de l'assignatura. | 2. Registra la situació. |
| | 3. Indica que vol realitzar un test. | 4. Registra la situació i genera una llista dels temes de l'assignatura. |
| | 5. Selecciona el tema. | 6. Registra la situació i genera una llista dels objectius del tema. |
| | 7. Selecciona l'objectiu. | 8. Genera un test d'acord a l'objectiu triat. |
| | 9. Resol les preguntes. | 10. Finalitza el test. |
| | 11. Visualitza el informe. | |
| Excepcions | 9a. S'acaba el temps. 9a1. Es dona el test per finalitzat. 9b. Té l'opció de visualitzar un recurs. 9b1. Cas d'ús: visualitzar un recurs (alumne com a actor). | |

3.3. Model de domini

A partir de tots els requisits de la nostra aplicació web i tota la funcionalitat que li haurem de donar que hem vist ens els anteriors capítols, ara podem dissenyar el nostre model de domini. El model de domini fixa les entitats que han de ser modelitzades dins l'aplicació i les interrelacions que hi ha entre elles.

Com aquest projecte tracta d'un mòdul de l'aplicació InnovaCampus, mostrarem el model de domini de tota l'aplicació InnovaCampus i afegirem una part ressaltada per mostrar quines parts s'han d'afegir per fer un disseny del meu mòdul.



Explicació de la part del model de domini que fa referència a les preguntes, les dependències, els recursos i els testos:

1. De l'entitat objectiu ens interessa saber que té atributs que correspon al número de preguntes màxim que pot tenir un test, a la puntuació de les respostes correctes i a la de les incorrectes. Els objectius estan compostos per una sèrie de preguntes, i d'aquesta manera s'obté la multiplicitat 1-N entre Objectiu i Pregunta.

2. Les preguntes també tenen un identificador únic i corresponen a un únic objectiu. Aquestes preguntes únicament tenen un atribut que es el text de la pregunta. Una pregunta o més d'una pot dependre d'una altra pregunta o de més d'una, establint així la relació de dependència entre preguntes. A més a més, una pregunta pot estar vinculada a un recurs, ja que per poder respondre correctament la pregunta necessitem la informació que ens dona un recurs. Per últim tenim que les preguntes tenen a la seva vegada un conjunt de respostes. Així obtenim la multiplicitat 1-N entre Pregunta i Resposta.

3. Els recursos tenen un identificador únic i corresponen a un únic objectiu. D'un recurs guardem el seu nom, el lloc on esta guardat i el tipus de recurs que es: imatge, vídeo, àudio o document. Un únic recurs pot estar associat a cap o moltes preguntes.

4. Les respostes s'identifiquen amb un identificador únic, i corresponen a una única pregunta. Tenen un atribut de text que és la resposta, un altre que indica si la resposta es correcta o no, i per últim un atribut que fa referència a un comentari que pot ficar el professor.

5. Un test està format per preguntes d'un mateix objectiu. Aquestes preguntes poden haver format part d'un altre test. D'aquí s'obté la multiplicitat M-N entre

Pregunta i Test. Quan un alumne decideix realitzar un test, es seleccionaran automàticament un sèrie de preguntes de l'objectiu seleccionat i es guardaran immediatament la data d'inici i el fet que encara estigui inacabat, així tindrem constància que el test encara no ha finalitzat. Un cop finalitzat el test s'indicarà la data de finalització. Un test pot tenir 3 estats de finalització: no finalitzat, finalitzat o acabat i fora de temps.

6. També desitgem tenir emmagatzemada la informació de la puntuació de cada test, per això hi ha la interrelació entre Test i Pregunta. A aquesta interrelació també hi hem afegit un atribut puntuació que correspon al valor numèric que ha obtingut l'alumne, que ha sol·licitat un test, a una pregunta específica, és a dir, la puntuació correcta o errònia de l'objectiu.

Capítol 4

Aspectes de disseny i implementació

En aquest capítol explicarem el disseny i la implementació del nostre mòdul pel projecte InnovaCampus.

Primer de tot explicarem el disseny de la base de dades, tant disseny de l'aplicació InnovaCampus com els canvis necessaris per poder acoblar el mòdul que estem dissenyant.

Ja que el projecte InnovaCampus i aquest mòdul estan dissenyats utilitzant el patró Model Vista Controlador, farem una breu explicació sobre aquest model i després nombrarem i explicarem les classes que formen part de cada mòdul del patró MVC.

També farem una descripció de l'algorisme de dependència entre preguntes i per acabar tractarem alguns aspectes de la implementació.

4.1. Disseny de la Base de Dades

A partir del model del domini obtingut en el capítol anterior, es pot fer un bon disseny d'una base de dades relacional. De la mateixa manera que en el model de domini, el disseny de la base de dades de l'aplicació InnovaCampus ja està dissenyat. En aquest cas, jo haure de fer una ampliació de la base de dades per dotar-la de les millores que estem fent.

Per facilitar la comprensió del disseny de la base de dades relacional que es mostra a continuació, s'han marcat en negreta els atributs referents a les claus primàries de cada entitat (l'atribut que els identifica) i les claus foranes (les que provenen d'una altra entitat i únicament podran tenir valors que estiguin inclosos en tals entitats) s'han subratllat.

Disseny de la base de dades actual d'InnovaCampus:

Usuari (**DNI**, Password, Email, Perfil)

Estudiant (**DNI**, Nom, Cognoms)

Professor (**DNI**, Nom, Cognoms, Telèfon)

Assignatura (**IdAssignatura**, Nom, Descripció)

Matriculat (**IdAssignatura**, **DNI**)

Responsable (**IdAssignatura**, **DNI**)

Tema (**IdTema**, Nom, Descripció, IdAssignatura)

Objectiu (**IdObjectiu**, Nom, Descripció, Número de preguntes, IdTema,
Puntuació Correcta, Puntuació Errònia)

Pregunta (**IdPregunta**, Text, IdObjectiu)

Resposta (**IdResposta**, Text, Correcta, IdPregunta, Comentari)

Test (**IdTest**, Nota, Hora inici, Hora fi, Estat final, DNI)

Resultat (**IdTest**, **IdPregunta**, Puntuació)

Anunci (**idAnunci**, tipus, text)

AnuncisAssignatura (**id**,text,idAssignatura)

Disseny de la base de dades adaptat a les noves millores:

(Totes les taules que no apareixen són les que no sofreixen cap canvi respecte al disseny anterior)

Pregunta (**IdPregunta**, Text, IdObjectiu, IdRecurs)

Recurs (**IdRecurs**, Tipus, Nom, Localitzacio, IdObjectiu)

Dependència (**IdPreguntaA**, **IdPreguntaB**)

4.2. Model Vista Controlador

El Model Vista Controlador (MVC) és un patró de disseny de software que separa les dades d'una aplicació, la interfície d'usuari, i la lògica de control en tres components diferents de forma que les modificacions al component de la vista poden ser realitzades amb un mínim impacte al component del model de dades. Això és útil ja que els models típicament tenen un cert grau d'estabilitat (depenent de l'estabilitat del domini del problema que està sent modelat), on el codi de la interfície d'usuari sigui més robust, degut a que el seu desenvolupador està menys propens a "trencar" el model mentre treballa de nou amb la vista.

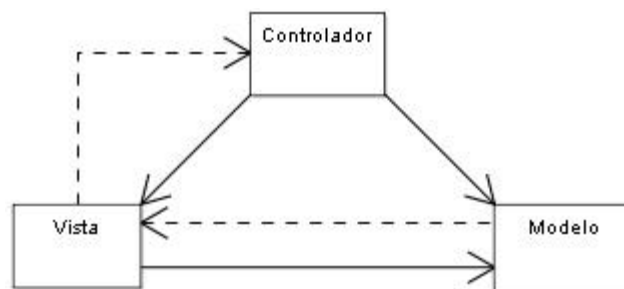
El patró MVC és molt freqüent en el món de la web, on la vista és la pàgina HTML i el codi que proveeix de dades dinàmiques a la pàgina. En termes generals, construir una aplicació utilitzant una arquitectura MVC implica definir tres classes de mòduls.

- **Model:** Aquesta és la representació específica del domini de la informació sobre el qual funciona l'aplicació. La lògica de domini assegura la integritat de les dades i permet derivar-ne noves. El model és una altra forma de cridar a la capa de domini.
- **Vista:** Aquest presenta el model en un format adequat per a interactuar, usualment un element de la interfície d'usuari.
- **Controlador:** Aquest respon als esdeveniments, usualment accions de l'usuari i invoca canvis en el model i probablement a la vista.

Encara que es pot torbar diferents implementacions de MVC, el flux que segueix el control generalment es el següent:

1. L'usuari interactua amb la interfície d'usuari d'alguna manera.
2. El controlador rep (per part dels objectes de la interfície-vista) la notificació de l'acció sol·licitada per l'usuari. El controlador gestiona l'esdeveniment que arriba, freqüentment a través de un gestor d'esdeveniments.
3. El controlador accedeix al model, actualitzant, possiblement modificant-lo de manera adequada a l'acció sol·licitada per l'usuari. Els controladors complexos estan molts cops estructurats utilitzant un patró de comanda que encapsula les accions i simplifica la seva extensió.
4. El controlador delega als objectes de la vista la tasca de desplegar la interfície d'usuari. La vista obté les seves dades del model per generar la interfície apropiada per l'usuari on es reflexa els canvis en el model. El model no ha de tenir coneixement directe sobre la vista. En algunes implementacions la vista no té accés directe al model, deixant que el controlador enviï les dades del model a la vista.
5. La interfície d'usuari espera noves interaccions de l'usuari, començant el cicle novament.

En el següent esquema podem observar gràficament la idea de l'arquitectura MVC:



4.3. Classes de disseny

En aquest apartat explicarem la funcionalitat de les diferents classes que formen part de la nostra aplicació.

Les classes estan dividides segons si formen part del Model, la Vista o el Controlador del Patró ModelVistaControlador. En les classes del Model hi ha tant les classes que formen part de l'aplicació InnovaCampus com les del meu mòdul. Les classes de la Vista i el Controlador són únicament les del meu mòdul.

4.3.1. Classes del Model

- **User.java**

Aquesta classe serveix per tenir una instància de l'entitat Usuari. Està formada pels atributs que formen part de l'entitat Usuari (DNI, Password, Email i Perfil). Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceUser.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de User. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de User.

- **Student.java**

Aquesta classe serveix per tenir una instància de l'entitat Estudiant. Està formada pels atributs que formen part de l'entitat Estudiant (Nom i Cognoms) i per un atribut (DNI) per saber amb quina instància de User està relacionat. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceStudent.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Student. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Student.

- **Teacher.java**

Aquesta classe serveix per tenir una instància de l'entitat Professor. Està formada pels atributs que formen part de l'entitat Professor (Nom, Cognoms i Telèfon) i per un atribut (DNI) per saber amb quina instància de User està relacionat. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceTeacher.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Teacher. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Teacher.

- **Register.java**

Aquesta classe serveix per tenir una instància de la relació de Matriculat entre un estudiant i una assignatura. Està formada per dos atributs (IdSubject, DNI), així s'estableix que l'alumne (DNI) està matriculat a l'assignatura (IdSubject). Els seus mètodes són únicament d'accés als seus atributs.

- **Service Register.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Register. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Register.

- **Responsible.java**

Aquesta classe serveix per tenir una instància de la relació de Responsable entre un professor i una assignatura. Està formada per dos atributs (IdSubject, DNI), així s'estableix que el professor (DNI) és el responsable de l'assignatura (IdSubject). Els seus mètodes són únicament d'accés als seus atributs.

- **Service Responsible.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Responsible. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Responsible.

- **Subject.java**

Aquesta classe serveix per tenir una instància de l'entitat Assignatura. Està formada pels atributs que formen part de l'entitat Assignatura (Id, Nom i Descripció). Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceSubject.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Subject. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Subject.

- **Topic.java**

Aquesta classe serveix per tenir una instància de l'entitat Tema. Està formada pels atributs que formen part de l'entitat Tema (Id, Nom i Descripció) i per un atribut (IdSubject) per saber amb quina instància de Subject està relacionat. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceTopic.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Topic. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Topic.

- **Objective.java**

Aquesta classe serveix per tenir una instància de l'entitat Objectiu. Està formada pels atributs que formen part de l'entitat Objectiu (Id, Nom, Descripció, NumeroPreguntes, PuntuacióCorrecta i PuntuacióErrònia) i per un atribut (IdTopic) per saber amb quina instància de Topic està relacionat. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceObjective.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Objective. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Objective.

- **Question.java**

Aquesta classe serveix per tenir una instància de l'entitat Pregutna. Està formada pels atributs que formen part de l'entitat Pregunta (Id i Text), per un atribut (IdObjective) per saber amb quina instància de Objective està relacionat i per un altre atribut (IdResource) per saber si la pregunta està associada o no amb un recurs i saber amb quina instància de Resource està relacionada. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceQuestion.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Question. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Question.

- **Answer.java**

Aquesta classe serveix per tenir una instància de l'entitat Resposta. Està formada pels atributs que formen part de l'entitat Resposta (Id, Text, Comentari i Correcta) i per un atribut (IdQuestion) per saber amb quina instància de Question està relacionat. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceAnswer.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Answer. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Answer.

- **Resource.java**

Aquesta classe serveix per tenir una instància de l'entitat Recurs. Està formada pels atributs que formen part de l'entitat Recurs (Id, Tipus, Nom i Localització) i per un atribut (IdObjective) per saber amb quina instància de Objective està relacionat. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceResource.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Resource. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Resource.

- **Dependency.java**

Aquesta classe serveix per tenir una instància de la relació de Dependència entre preguntes. Està formada per dos atributs (IdQuestionA, IdQuestionB), així s'estableix que la pregunta B depèn de la pregunta A. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceDependency.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Dependency. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Dependency.

- **Result.java**

Aquesta classe serveix per tenir una instància de l'entitat Resultat. Està formada pels atributs que formen part de l'entitat Resultat (Puntuació) i per dos atributs (IdTest, IdQuestion) per saber amb quines instàncies de Test i Question està relacionat. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceResult.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Result. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Result.

- **Test.java**

Aquesta classe serveix per tenir una instància de l'entitat Test. Està formada pels atributs que formen part de l'entitat Test (Id, Nota, HoraInici, HoraFinal i EstatFinal) i per un atribut (DNI) per saber amb quina instància de Student està relacionat. Els seus mètodes són únicament d'accés als seus atributs.

- **ServiceTest.java**

Aquesta classe serveix per connectar la informació persistent que tenim en la base de dades amb informació que es troba en mode d'execució d'una o més instàncies de Test. Crea, obté, modifica o elimina la informació de la base de dades sobre les instàncies de Test.

- **TestGenerator.java**

Aquesta classe serveix per generar tests. A partir d'un objectiu donat, genera un test de preguntes aleatòries tenint en compte la possible dependència entre preguntes.

- **boolIdText.java**

Aquesta classe està formada per tres atributs (Booleà, Identificador i Text). En l'aplicació aquesta classe s'utilitza per relacionar-la amb una instància de Dependency i així poder saber si es vol que existeixi o no la dependència de la pregunta que estem creant o modificant a la pregunta de l'instància relacionada.

4.3.1. Classes de la Vista

- **newQuestionForm.java**

Aquesta classe serveix per capturar o mostrar les dades que introdueix o visualitza l'usuari. En aquest cas les dades estan relacionades amb la creació, modificació, visualització o eliminació d'una pregunta. Això inclou les dades de la pregunta, les respostes d'aquesta pregunta, si esta o no associat a un recurs i si te alguna dependència a una altra pregunta.

- **newResourceForm.java**

Aquesta classe serveix per capturar o mostrar les dades que introdueix o visualitza l'usuari. En aquest cas les dades estan relacionades amb la creació, modificació, visualització o eliminació d'un recurs.

- **listQuestionForm.java**

Aquesta classe serveix per mostrar una llista de preguntes d'un objectiu determinat amb la intenció de que l'usuari triï la pregunta que desitja per fer una acció escollida prèviament.

- **listResourceForm.java**

Aquesta classe serveix per mostrar una llista de recursos d'un objectiu determinat amb la intenció de que l'usuari triï el recurs que desitja per fer una acció escollida prèviament.

- **resourceForm.java**

Aquesta classe serveix per obtenir tota la informació necessària per poder visualitzar un recurs determinat prèviament i les seves dades.

- **nextQuestionForm.java**

Aquesta classe serveix per mostrar les preguntes del test a l'usuari i capturar les seves respostes.

- **newQuestion.jsp**

Aquesta pàgina serveix per emplenar les dades per crear una nova pregunta i per modificar les dades d'una pregunta ja creada.

- **viewQuestion.jsp**

Aquesta pàgina serveix per visualitzar tota la informació referent a una pregunta determinada.

- **listQuestion.jsp**

Aquesta pàgina serveix per llistar les preguntes associades a un objectiu, escollir la que es vulgui i per poder-la modificar o visualitzar.

- **listQuestionToRemove.jsp**

Aquesta pàgina serveix per llistar les preguntes associades a un recurs i escollir la que es vol eliminar.

- **newResource.jsp**

Aquesta pàgina serveix per crear un nou recurs, ja sigui imatge, vídeo, àudio o un document.

- **listResourceToView.jsp**

Aquesta pàgina serveix per llistar els recursos associats a un objectiu determinat, escollir el que es vulgui i per poder-lo visualitzar.

- **listResourceToEdit.jsp**

Aquesta pàgina serveix per llistar els recursos associats a un objectiu determinat, escollir el que es vulgui i per poder-lo modificar.

- **editResource.jsp**

Aquesta pàgina serveix per modificar un recurs, tant pot ser el seu nom, com el fitxer o el tipus de fitxer.

- **listResourceToRemove.jsp**

Aquesta pàgina serveix per llistar els recursos associats a un objectiu determinat, escollir el que es vulgui i per poder-lo eliminar.

- **Question.jsp**

Aquesta pàgina serveix per poder veure una pregunta del test que s'està realitzant i triar la resposta que es cregui correcta.

- **resultTest.jsp**

Aquesta pàgina serveix per mostrar el resultat d'un test finalitzat. Es mostra la puntuació, les respostes correctes i les errònies.

- **Imatge.jsp**

Aquesta pàgina serveix per poder visualitzar un recurs del tipus imatge.

- **Video.jsp**

Aquesta pàgina serveix per poder visualitzar un recurs del tipus vídeo.

- **Audio.jsp**

Aquesta pàgina serveix per poder visualitzar un recurs del tipus àudio.

- **Document.jsp**

Aquesta pàgina serveix per poder visualitzar un recurs del tipus document.

4.3.1. Classes del Controlador

- **newQuestionAction.java**

Aquesta classe serveix per capturar l'esdeveniment de crear una nova pregunta i obtenir tota la informació que l'usuari visualitzarà, com els possibles recurs i les possibles dependències.

- **addQuestionAction.java**

Aquesta classe serveix per processar les dades relacionades amb una pregunta, amb les seves respostes, possible recurs i possibles relacions de dependència., i crear o modificar la pregunta.

- **listQuestionAction.java**

Aquesta classe serveix per capturar l'esdeveniment de mostrar una llista de preguntes i preparar la llista d'acord a les preguntes que formen part de l'objectiu triar per l'usuari.

- **viewQuestionAction.java**

Aquesta classe serveix per capturar l'esdeveniment de visualitzar una pregunta i obtenir tota la informació relacionada amb la pregunta escollida per l'usuari.

- **removeQuestionAction.java**

Aquesta classe serveix per capturar l'esdeveniment d'eliminar una pregunta i elimina la pregunta.

- **newResourceAction.java**

Aquesta classe serveix per capturar l'esdeveniment de crear un nou recurs i amb les dades capturades crear aquest nou recurs i que formi part de l'aplicació.

- **prepareEditResourceAction.java**

Aquesta classe serveix per capturar l'esdeveniment de modificar un recurs determinat.

- **editResourceAction.java**

Aquesta classe serveix per modificar un recurs de la base de dades. Ja sigui el nom, el tipus de fitxer o el fitxer font.

- **list ResourceAction.java**

Aquesta classe serveix per capturar l'esdeveniment de mostrar una llista de recursos i preparar la llista d'acord als recursos que formen part de l'objectiu triar per l'usuari.

- **viewResourceAction.java**

Aquesta classe serveix per capturar l'esdeveniment de visualitzar un recurs i obtenir tota la informació relacionada amb el recurs per poder saber si quin tipus de recurs s'ha de visualitzar.

- **removeResourceAction.java**

Aquesta classe serveix per capturar l'esdeveniment d'eliminar un recurs i eliminar-lo.

- **generarTestAction.java**

Aquesta classe serveix per capturar l'esdeveniment de voler realitzar un test, i preparar les preguntes que tindrà el test d'acord a l'objectiu triat.

- **nextQuestionAction.java**

Aquesta classe serveix per preparar la informació necessària referent a la següent pregunta del test. En cas que no quedin més preguntes es passarà a la finalització del test.

- **resultTestAction.java**

Aquesta classe serveix per quan s'ha acabat un test calcular la nota del test i guardar els resultats.

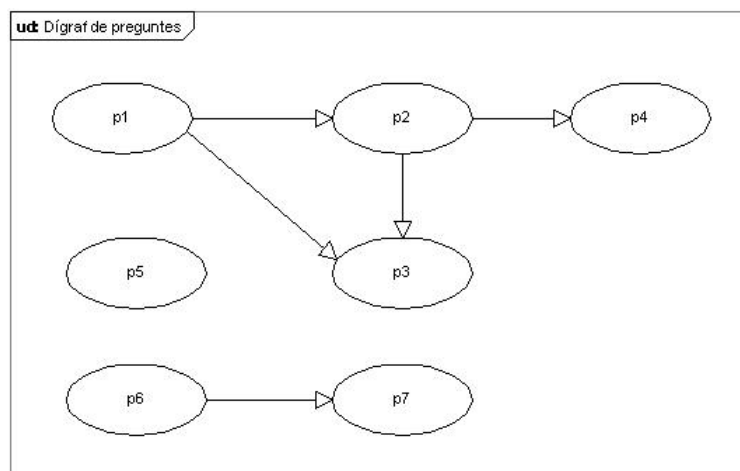
4.4. Algorisme de dependència entre preguntes

Aquest algorisme està dissenyat per crear la relació de dependència entre preguntes i poder generar test de manera aleatòria tenint en compte la dependència entre les preguntes.

4.4.1. Idea bàsica

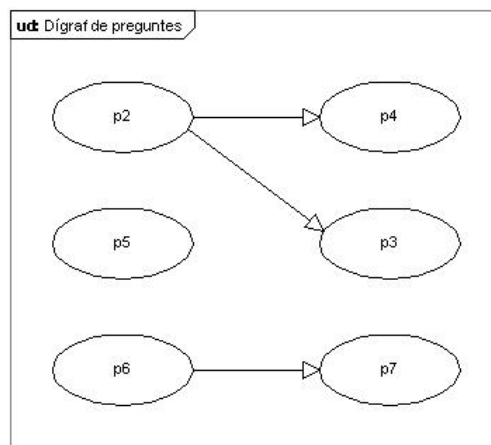
Quan diem que una pregunta B depèn d'una altra pregunta A, ens referim a que la pregunta B no pot sortir al mateix test si abans no ha sortit la pregunta A. En aquest cas és fàcil veure la relació que hi ha entre preguntes i l'ordre en que poden ser escollides, però quan tenim un nombre més gran de preguntes i més relacions de dependència, la millor manera per representar aquestes relacions és mitjançant un dígraf (graf dirigit).

Per exemple, tenim el conjunt de preguntes {p1, p2, p3, p4, p5, p6, p7} i les següents relacions de dependència: p2 depèn de p1, p3 depèn de p1 i p2, p4 depèn de p2, i p7 depèn de p6.



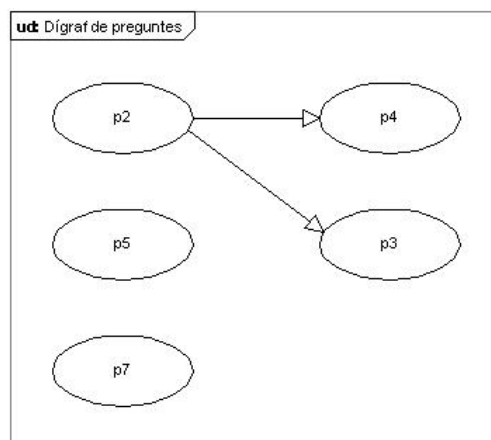
Visualitzant la representació gràfica es fàcil veure que les úniques preguntes que poden ser escollides en la primera opció són les que no depenen de cap altra pregunta, és a dir, el conjunt format per $\{p1, p5, p6\}$

Si escollim per exemple la pregunta p1, com aquesta ja ha set escollida la trauríem del dígraf, juntament amb totes les seves dependències, i ens quedaria així:

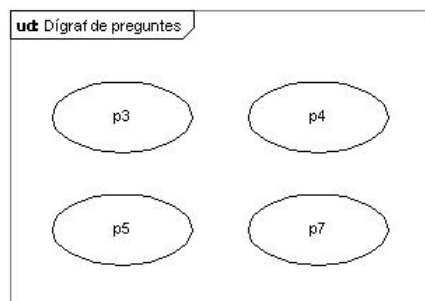


Ara es pot triar entre el conjunt de preguntes $\{p2, p5, p6\}$

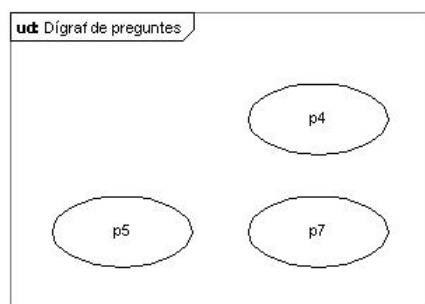
Si escollim la pregunta p6, el dígraf quedaria de la següent forma:



Ara el conjunt de preguntes a triar es $\{p2, p5, p7\}$ i fins al moment el test estaria format per les preguntes $\{p1, p6\}$. Si escollim l'opció de la pregunta p2 ens queda:



Ara podem triar entre $\{p4, p3, p5, p7\}$, escollim la pregunta p3 i el dígraf resultant es el següent:



Imaginem que el test es de 5 preguntes i ja en portem quatre, per tant com a ultima pregunta escollim la p5.

El test resultant es $\{p1, p6, p2, p3, p5\}$. Si imaginem que les preguntes escollides han set a l'atzar, tenim que hem aconseguit generar un test aleatori tenint en compte les dependències entre preguntes.

El procediment que acabem de descriure es pot esquematitzar de la següent manera:

(Anar agafant preguntes del dígraf i afegint-les al test.)

acció resolucióDependencia (d:dígraf , n:natural) : llista:vector[1..n] de preguntes

 llista: vector[1..n] de preguntes

 i:=0

mentre i<n *fer*

 // afegir una pregunta a la llista

 llista.afegir (generarPregunta(dígraf))

 i:=i+1

fmentre

facció

(Escull a l'atzar una pregunta que tingui el node lliure. El node seleccionat i totes les seves dependències s'eliminen.)

acció generarPregunta (d:e/s dígraf) : p:pregunta

 p : pregunta

 aleatori : natural

 // obtenim les preguntes lliures, que no depenen de cap altra pregunta

 llistaPosiblesPreguntes := d.obtenirNodesLliures ()

 // creem un número aleatori

 aleatori := random (llistaPosiblesPreguntes.longitud)

 p := llistaPosiblesPreguntes [aleatori]

 // eliminem el node de la pregunta escollida. També s'eliminaran les seves dependències.

 d.eliminarNode (p)

 retorna p

facció

4.4.2. Integració en l'aplicació

Per generar els tests en l'aplicació hem fet servir la classe TestGenerator. El disseny d'aquesta classe esta pensat en l'exemple anterior. A partir d'un identificador d'objectiu la classe genera un dígraf de totes les preguntes de l'objectiu determinat i les seves relacions de dependència.

Per poder treballar amb un dígraf (graf dirigit) hem fet servir la llibreria JGraphT, que es una llibreria lliure de grafs de Java que proporciona d'objectes i algorismes matemàtics de la teoria de grafs. Per més informació sobre aquesta llibreria vegeu l'annex C: JGraphT.

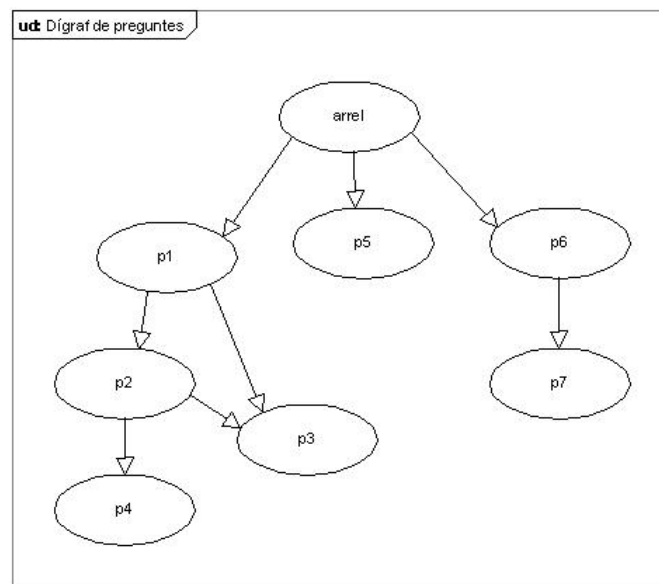
Per facilitar la funcionalitat, es crea un node anomenat arrel on els seus subnodes són els nodes que tenen les preguntes a escollir.

Per generar el test es segueixen els següents passos:

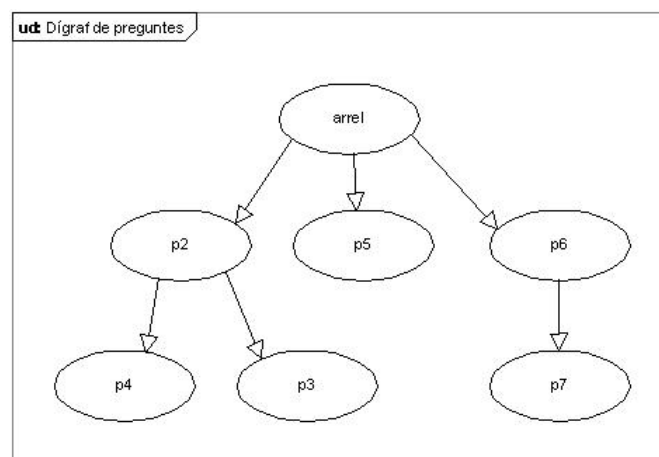
1. S'elegeix a l'atzar una pregunta que depengui directament del node arrel.
2. El node seleccionat s'ha d'eliminar, i en els subnodes poden passar dos coses. Si depenen única i exclusivament del node elegit, ara passen a dependre del node arrel, i per tant són preguntes que poden ser escollides. Els altres nodes que depenien tant del node elegit com d'altres nodes, continuen depenent del altres nodes i s'hauran d'esperar més torns per poder ser escollides.
3. Es tornar al pas 1 fins a completar el nombre de preguntes del test o fins que no hi ha cap més pregunta.

Representació gràfica de l'evolució del dígraf de l'aplicació seguint l'exemple anterior.

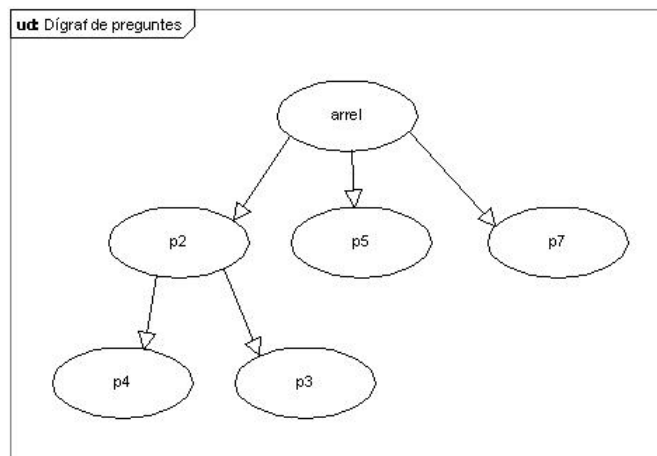
Conjunt de preguntes {p1, p2, p3, p4, p5, p6, p7}



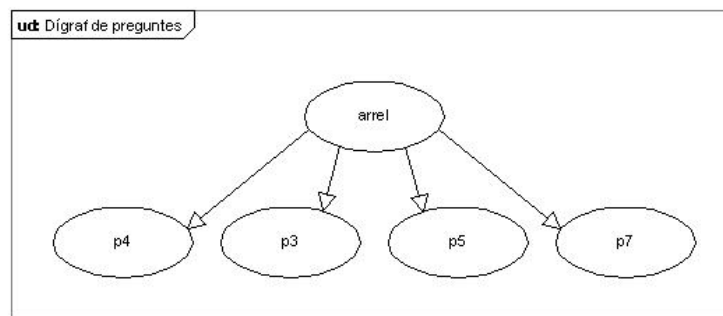
Escollim la pregunta p1



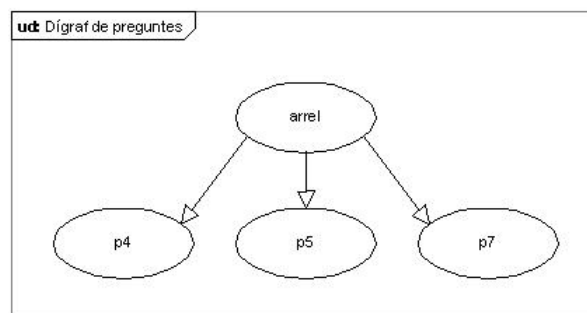
Escollim la pregunta p6



Escollim la pregunta p2



Escollim la pregunta p3



Per acabar escollim la pregunta p5

El test estarà format per les preguntes {p1, p6, p2, p3, p5}

El procediment que acabem de descriure es pot esquematitzar de la següent manera:

```
private DirectedGraph<Question,DefaultEdge> dg =
    new SimpleDirectedGraph<Question,DefaultEdge>(DefaultEdge.class);
private Question rootQ;
private List<Question> list = new List<Question>;

public void generateTest ( ) {
    Question q;
    for ( int i=0 ; i<numeroPreguntesTest; i++ ) {
        q = generateQuestion ( );
        list.add ( q );
    }
}

private Question generateQuestion ( ) {
    // guardem en una col·lecció totes les arestes que surten de la pregunta rootQ (arrel)
    Set<DefaultEdge> edges=dg.outgoingEdgesOf(rootQ);
    // creem un iterador d'aquesta col·lecció que acabem de fer
    Iterator<DefaultEdge> it= edges.iterator();
    Random r=new Random();
    Question q=null;
    // si la col·lecció te algun element
    if(edges.size()!=0){
        // creem un número aleatori entre 0 i la llargada de la col·lecció
        int aleatori=r.nextInt(edges.size());
        // movem l'iterador fins la posició del número aleatori
        for(int i=0;i<aleatori;i++) it.next();
        // agafem la pregunta apuntada per l'aresta marcada per l'iterador
        q=dg.getEdgeTarget(it.next());
        // eliminem la pregunta i les seves dependències del digraf
        deleteQuestion(q);
    }
    return q;
}
```

4.5. Aspectes de la implementació

En aquest apartat, explicarem breument alguns dels aspectes més importants en la implementació del nostre mòdul.

Durant tota l'estona hem estat parlant de la possibilitat d'associar un recurs a una pregunta, i que els recursos possibles podrien ser imatges, vídeos, àudios i documents. Per poder visualitzar una imatge no hi ha cap problema, pels documents tampoc es un gran problema ja que la majoria de documents serien en pdf i es poden visualitzar a partir del navegador sense cap dificultat. El tipus de recurs de documents, permet associar qualsevol tipus d'arxiu, de manera que si no es pugues visualitzar amb el navegador podries baixar-lo i visualitza'l amb el programa adequat. Pel que fa als tipus de recurs de vídeo i àudio hem tingut més problemes. Els arxius d'aquest tipus que pots visualitzar per Internet normalment es a partir de flash o real player, windows media player, quicktime... Hem decidit utilitzar flash perquè es el mètode més extens per visualitzar arxius de vídeo per la xarxa. Els altres mètodes no els considerem tant bons ja que exigeixen que l'usuari tingui instal·lat un software determinat dins l'ordinador. En canvi el flash disposa d'un pluguin que és molt senzill d'instal·lar. A més a més, molts usuaris que naveguen per Internet sovint, ja el tenen instal·lat. Per aquestes raons hem decidit que la visualització dels vídeos i la reproducció dels àudios es faci per mitja de reproductors flash. Per més informació sobre els reproductors flash player consultar [9] i [10].

Un altre problema relacionat amb el que acabem de comentar es que la visualització dels vídeos amb el reproductor flash exigien tenir els vídeos en un format determinat (flv). Per poder passar del tipus d'arxiu que pujava el professor

(ja sigui avi, mpg o altres) a flv, hem tingut de fer servir una col·lecció de software lliure anomenada ffmpeg que pot fer aquesta conversió d'arxius (avi, mpeg...) a flv. Fem que quan es pugi al servidor un arxiu vídeo es converteixi a un arxiu flv. Per més informació sobre el programa ffmpeg consultar [13].

Un altre dels aspectes a tractar en la implementació del nostre mòdul és la utilització de javascript i css. Aquests dos llenguatges ens han permès poder donar-li a les pàgines web un aspecte més bonic i donar-li un major dinamisme. Javascript ho utilitzem més per amagar o ensenyar part de la pàgina, per controlar la temporització en els tests, per redimensionar finestres, etc. i el llenguatge css el fem servir per donar més forma i color a la pàgina.

Capítol 5

Conclusió i treball futur

Primer de tot m'agradaria parlar sobre la tecnologia Jakarta-Struts utilitzada per fer el mòdul de l'aplicació. Em va suposar un gran esforç entendre el seu funcionament. Entenia cada part de patró MVC per separat i entenia les classes i la funcionalitat que tenia, però em va costar una mica més adonar-me i entendre com es connecten entre si aquestes tres parts. Considero que Jakarta-Struts és una tecnologia nova i molt interessant. Amb tots aquests mesos que he estat treballant, he après de manera autodidacta a entendre i utilitzar aquesta tecnologia. He fet aquesta aplicació i considero que està ben feta, però hi ha parts de la tecnologia struts que no l'he inclòs en l'aplicació perquè no estava relacionat o no es podia utilitzar. En un treball futur i de manera personal m'agradaria aprofundir en aquesta tecnologia i poder entendre i saber fer servir correctament tota la seva funcionalitat.

A més a més d'assolir el coneixement d'una nova tecnologia, aquest treball m'ha servit per tenir més experiència en el treball en grup i en el procediment a seguir

per dissenyar i desenvolupar aplicacions. Considero que abans de començar a programar primer de tot s'ha de saber el funcionament de la tecnologia a utilitzar i s'ha de pensar bé quina es la funcionalitat que es vol donar a l'aplicació.

Un dels aspectes que considero esmentar es el software lliure. Per fer aquest projecte he fet ús de diverses llibreries lliures que he trobat per la xarxa. Aquestes llibreries m'han ajudat molt en el desenvolupament de l'aplicació i m'han evitat dissenyar i implementar unes classes necessàries. Crec que el software lliure és important, ja que es una manera d'ajudar als altres fent una cosa que a un mateix li agrada.

Aquest mòdul que he dissenyat i implementat, només es el començament d'un seguit de canvis que actualment s'estan fent. Hi ha un altre mòdul per millorar les estadístiques que un professor pot veure respecte als tests realitzats i el progrés dels alumnes, i també s'està actualitzant tota l'aplicació per actualitzar-la a la tecnologia struts (menys les dos parts dels mòduls que ja es fa directament en struts). Aquests només son uns canvis que s'estan fent ara, però com a treball futur podríem afegir les següents funcionalitats a l'aplicació:

- Poder fer tests relacionats amb un recurs. Generar un test de preguntes, on totes les preguntes estiguin associades a un recurs determinat, ja sigui imatge, vídeo, àudio o un document.
- Poder fer tests d'un tema o assignatura determinada, fent que el test es generi a partir de totes les preguntes possibles.
- Inserció de preguntes a partir de fitxer. Generar un mòdul que permeti incloure un conjunt de preguntes a l'assignatura, tema o objectiu a partir d'un document de text. En aquest document de text estarien escrites les preguntes amb les respostes i tota la informació associada a la pregunta i el

mòdul s'encarregaria de llegir-les i emmagatzemar-les. Per més informació consultar l'annex E: Preguntes des de fitxer.

- Establir alguna altra relació entre les preguntes, com preguntes excloents, és a dir, que si surt una pregunta, l'altra no pot sortir perquè tracten del mateix aspecte o son semblants.

Per finalitzar, m'agradaria dir que m'ha agradat poder formar part del projecte InnovaCampus. Crec que és un projecte important que només utilitza un o dos professors. Desitjo que el seu ús s'estengui molt més i sigui una eina més per l'estudiant com ho és el Campus Virtual (sakai).

Annex A

Tecnologies web

Com hem parlat anteriorment, InnovaCampus és un portal web. Per poder entendre com estan creades les pàgines web que formen part d'aquest portal hem de parlar de diverses tecnologies com html, javascript, css, jsp, llenguatges utilitzades principalment en pàgines web.

A.1. HTML

HTML (Acrònim de Hyper Text Markup Language, "llenguatge de marcat d'hipertext"), és un llenguatge de marcat que deriva de l'SGML dissenyat per estructurar textos i relacionar-los en forma d'hipertext. Gràcies a Internet i als navegadors web, s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents.

Aquesta tecnologia utilitza etiquetes o marques, que consisteixen en breus instruccions de començament i final, mitjançant les quals es determina la forma en la que han d'aparèixer en el navegador el text, així com també les imatges i el altres elements, en la pantalla de l'ordinador.

Tota etiqueta s'identifica perquè està tancada entre els signes menor que i major que (< >), i algunes tenen atributs que poden prendre algun valor.

En general les etiquetes s'aplicaran de dos formes especials:

- S'obrin i es tanquin, com per exemple: `negreta` que es veuria en el navegador com **negreta**.
- No poden obrir-se i tancar-se, com `<hr/>` que es veuria en el navegador com una línia horitzontal.

Les etiquetes bàsiques d'HTML, d'obligada presència en tot document són:

- `<!DOCTYPE>`: És l'etiqueta que permet definir el tipus de document HTML que s'emptra. Existeixen tres tipus bàsics: l'estricta (Strict), el transicional (Transitional) i el de marcs (Frameset).
- `<html>`: És l'etiqueta arrel de qualsevol document HTML o XHTML.
- `<head>`: Defineix la capçalera del document HTML. Permet declarar metainformació del document que no es mostra directament en el navegador. Aquesta informació és d'especial rellevància pels indexadors i cercadors automàtics.
- `<body>`: Defineix el cos del document. Aquesta és la part del document HTML que es mostra en el navegador.

Dintre de la capçalera `<head>` hi podem trobar:

- `<title>`: Permet definir el títol de la pàgina. En navegadors gràfics el contingut del títol apareix a la barra del títol a sobre de la finestra.

- `<meta>`: Permet definir metainformacions del document tals com l'autor, la data de realització, la codificació del document (UTF, ISO, etc.), les paraules clau i la descripció del mateix
- `<link>`: Permet definir metadades complementàries a les del meta tals com el document anterior, el següent, el capítol al qual pertany el document, la pàgina glossari, etc.

Dintre del cos `<body>` hi podem trobar:

- `<a>`: Etiqueta ancora. Crea un enllaç a un altre document o a una altra zona del mateix, segons els atributs.
- `<h1>`, `<h2>`,... `<h6>`: capçaleres o títols del document, acostumen a distingir-se per mida.
- `<div>`: Divisió estructural de la pàgina.
- `<p>`: Paràgraf.
- `
`: Salt de línia.
- `<table>`: Indica el començament d'una taula, després s'haurà de definir les files amb `<tr>` i les cel·les dintre de les files amb `<td>`.
- ``: Llista desordenada (sense numerar). Els ítems es defineixen amb ``.
- ``: Llista ordenada (numerat). Els ítems es defineixen amb ``.
- `<dl>`: Llista de definició. Hi ha dos tipus d'ítem; el `dt` i el `dd`.
 - `<dt>`: Terme a definir.
 - `<dd>`: Definició del terme.

Per més informació consultar [5] i [17].

A.2. JavaScript

JavaScript és un llenguatge script basat en el concepte de prototip, implementat originàriament per Netscape Communications Corporation, i que va derivar en l'estàndard ECMAScript. Tots els navegadors interpreten el codi JavaScript integrat dins de les pàgines web.

JavaScript és un llenguatge interpretat, és a dir, no requereix compilació. El navegador de l'usuari s'encarrega d'interpretar les sentències JavaScript contingudes en una pàgina HTML i executar-les adequadament. Es pot incloure en qualsevol document HTML o tot aquell que s'acabi traduïnt en HTML en el navegador del client, ja sigui PHP, ASP, SVG, JSP ...

El codi va inscrit dins dels elements HTML `<script>` i `</script>`:

```
<script type="text/javascript">  
    // código JavaScript  
</script>
```

JavaScript és un llenguatge orientat a esdeveniments. Quan un usuari clica sobre un enllaç o mou el punter sobre una imatge es produeix un esdeveniment. Mitjançant JavaScript es poden desenvolupar scripts que executin accions com a resposta a aquests esdeveniments.

Característiques tècniques de JavaScript.

- Interpretat (no compilat) des de el navegador del client.
- Enllaçat dinàmic. Els objectes als que es fa referència han d'existir en temps d'execució (llenguatge interpretat).

- No requereix la presència de servidor, ja que manté un funcionament local.
- L'aplicació resideix en el client.
- El codi està integrat en el HTML. Queda ocult a l'usuari , però pot formar part del mateix arxiu.
- Basat en objectes. Només els utilitza, manca d'herències i de classes.
- No és necessari declarar el tipus de variables.

Per més informació consultar [3] i [17].

A.3. CSS

Els fulls d'estil en cascada (Cascading Style Sheets, CSS) són un llenguatge formal usat per a definir la presentació d'un document estructurat escrit en HTML o XML (i per extensió en XHTML). El W3C (World Wide Web Consortium) és l'encarregat de formular l'especificació dels fulls d'estil que servirà d'estàndard per als agents d'usuari o navegadors.

La idea que es troba darrere del desenvolupament de CSS és separar l'estructura d'un document de la seva presentació.

Per exemple, l'element d'HTML `<H1>` indica que un bloc de text és un encapçalament i que és més important que un bloc etiquetat com `<H2>`. Versions més antigues d'HTML permetien atributs extra dins l'etiqueta oberta per donar-li format (com el color o la mida de font). No obstant això, cada etiqueta havia de disposar d'aquesta informació si es desitjava un disseny consistent per a una pàgina, i a més, una persona que llegís aquesta pàgina amb un navegador, perdia totalment el control sobre la visualització del text.

Quan s'utilitza CSS, l'etiqueta `<H1>` no hauria de proporcionar informació sobre com serà visualitzat, només marca l'estructura del document. La informació d'estil separada en una fulla d'estil específica com s'ha de mostrar `<H1>`: color, font, alineació del text, grandària, i altres característiques no visuals com definir el volum d'un sintetitzador de veu, per exemple.

La informació d'estil pot ser adjuntada tant com un document separat o en el mateix document HTML. En aquest document, podrien definir-se estils generals en la capçalera del document o en cada etiqueta particular mitjançant l'atribut "style".

Els avantatges d'utilitzar CSS (o altre llenguatge d'estil) són:

- Control centralitzat de la presentació d'un lloc web complet amb el que s'agilita de forma considerable l'actualització del mateix.
- Els navegadors permeten als usuaris especificar el seu propi full d'estil local que serà aplicada a un lloc web remot, amb el que augmenta considerablement l'accessibilitat. Per exemple, persones amb deficiències visuals poden configurar el seu propi full d'estil per a augmentar la grandària del text o remarcar més els enllaços.
- Una pàgina pot disposar de diferents fulles d'estil segons el dispositiu que la mostri o fins i tot a elecció de l'usuari. Per exemple, per a ser impresa, mostrada en un dispositiu mòbil, o ser "llegida" per un sintetitzador de veu.
- El document HTML en si mateix és més clar d'entendre i s'aconsegueix reduir considerablement la seva grandària.

Els dissenyadors web van optar per la utilització de taules per al disseny de pàgines webs. D'aquesta forma, l'ús que donaven a les taules era com elements d'estil quan la seva veritable comesa és albergar contingut. La maquetació CSS consisteix a utilitzar capes (layers) que en HTML es defineixen amb les marques `<div></div>` (d'obertura i tancament, respectivament) per a estructurar l'aplicació web, com alternativa a les taules i els marcs (frames). Així, podem anomenar cada capa amb l'atribut `id` de manera que podrem definir les propietats d'aquesta capa en l'arxiu CSS extern al document, encara que també es pot incloure el codi CSS en la mateixa pàgina que l'HTML, però és poc funcional (és convenient fer ús de la modularitat per a fer més senzill el manteniment). Un exemple d'utilització de la marca `div` seria:

Exemple.html

```
dins de body
<div id="capa1">
    Aquí podem posar el que volguem: imatges, text, etc.<br>
</div>
```

Estil_Exemple.css

```
#capa1 {
    /*aquí definim les propietats de la capa*/
    background-color: #fff;
    margin: auto;
    color: #000;
}
```

Per més informació consultar [14] i [17].

A.4. JSP

JavaServerPages (JSP) és una tecnologia basada en Java que permet generar contingut dinàmic per pàgines webs, en forma de documents HTML, XML o altres.

Les pàgines JSP permeten tant la utilització de porcions de codi Java incrustades, com utilitzar algunes accions JSP predefinides mitjançant etiquetes. Aquestes etiquetes poden ser enriquides utilitzant Llibreries de Etiquetes (TagLibs o Tag Libraries) externes i inclús personalitzades.

D'aquesta manera, quan una pàgina és sol·licitada per un usuari i processada per un servidor HTTP, el codi HTML passarà directament a l'usuari, mentre que les porcions de codi Java seran executades al servidor per tal de generar el contingut dinàmic de la pàgina. Quan l'usuari accedeixi al codi de la pàgina només veurà HTML, sense poder accedir, per tant, al codi JSP subjacent.

La majoria de les implementacions de JSP estan basades en els Servlets. Els Servlets són programes basats en Java implementats mitjançant un contenidor de Servlets associats a un servidor HTTP. El fonament dels Servlets és el següent: un conjunt de URLs són configurades per a ser administrades per el contenidor de Servlets, de manera que sempre que arribi una sol·licitud per a una d'aquestes URLs al servidor, aquest l'envia al contenidor de Servlets per a que el processi.

La forma d'enviar-lo és creant un objecte Java que empaqueti totes les dades de la sol·licitud. Un objecte Java també és creat representant la resposta. Ambdós objectes tindran els seus mètodes d'accés, d'aquesta forma, el contenidor de Servlets accedeix a les dades de la sol·licitud per tal de realitzar les operacions

necessàries sobre els mateixos i així construir la resposta. El codi HTML generat com a resposta (no s'ha d'oblidar que el contingut dinàmic generat és codi HTML) és escrit a la cadena de sortida associada a l'objecte resposta, i aquest objecte és enviat al servidor HTTP, el qual la retorna al navegador que havia fet la sol·licitud en primer lloc. En el cas que existeixin múltiples sol·licituds per a un Servlet, aquestes són administrades executant cada crida als mètodes dels Servlets en diferents fils d'execució.

El component principal d'una implementació de JSP basada en Servlets és un Servlet especial anomenat compilador de pàgina. El contenidor està configurat per a cridar a aquest Servlet sempre que arribi una sol·licitud a una pàgina JSP. És aquest compilador de pàgina i la seva classe Java associada el que converteix el contenidor Servlet en un contenidor JSP.

El procediment és el següent: Quan arriba al servidor HTTP una sol·licitud d'una pàgina JSP, aquesta és enviada al contenidor JSP, el qual invoca al compilador de pàgina per a que s'encarregui de la mateixa. El compilador analitza el contingut de la pàgina buscant etiquetes JSP, traduint el seu contingut en el codi Java equivalent que, al ser executat, generarà el contingut dinàmic. Mesclant el contingut estàtic de la pàgina original junt amb el codi Java del contingut dinàmic, es generarà un Servlet amb els seus mètodes de servei. Una vegada que tot el codi del Servlet hagi estat construït, el compilador de la pàgina crida al compilador Java per a compilar aquest codi i afegir el fitxer de classe Java resultant al directori apropiat en el path de les classes del contenidor JSP. Tot aquest procés només es realitza la primera vegada que es sol·licita una pàgina JSP, la resta de sol·licituds són remeses directament al Servlet compilat. Així quan es crida a una pàgina JSP el compilador de pàgina invoca a aquest Servlet per tal de generar la resposta per a la sol·licitud original.

Resumint, podriem dir que les sol·licituds del navegador arriben al servidor HTTP i les pàgines JSP son enviades al Servlet compilador de pàgines que corre al contenidor JSP. Si el Servlet per a la pàgina actual està actualitzat, el genera i compila, carregant-lo al contenidor de Servlets. En el cas contrari el control és transferit al Servlet de la pàgina JSP que s'encarrega de manejar la sol·licitud generant la resposta i enviant-la al servidor HTTP el qual la remetrà al navegador.

Per més informació consultar [15].

Annex B

Jakarta-Struts

Struts és una eina de suport per al desenvolupament d'aplicacions web sota el patró MVC.

El marc de treball (o framework) de codi obert Struts es va crear per permetre que els desenvolupadors poguessin crear aplicacions web basant-se en les tecnologies Java Servlet i JavaServer Pages (JSP). Struts proporciona als desenvolupadors una infraestructura unificada sobre la que es pot basar les aplicacions d'Internet. Utilitzar Struts com la base, permet als desenvolupadors centrar-se en la creació de l'aplicació del negoci enlloc de centrar-se en la infraestructura.

El patró d'arquitectura MVC no està directament relacionat amb les aplicacions web o Java. De fet, es bastant comú en aplicacions Smalltalk que, per lo general no tenen res a veure amb la web.

El patró MVC té tres components clau:

- Components del Modelo.

Correspon a la lògica del negoci amb la que es comunica l'aplicació web. Normalment el model compren accessos a base de dades o sistemes que funcionen independentment de l'aplicació web.

- Components de la Vista.

Responsable d'una vista de presentació de la lògica de negoci. Consten pel general de Form beans i pàgines HTML i JSP. Les pàgines HTML s'utilitzen per servir contingut estàtic, mentre que les pàgines JSP se poden utilitzar per servir contingut tant estàtic om dinàmic. Els Form beans són contenidors de dades que proveeixen el camí per transferir dades entre la vista i el model. Contenen informació d'un formulari html que s'enviarà al model o informació del model que s'enviarà a la vista.

- Components del Controlador.

Responsable de controlar el flux i estat de l'entrada de dades per part de l'usuari. El controlador d'una aplicació web realitza les següents funcions:

1. Interceptar peticions HTTP des d'un client.
2. Traduir cada petició en una operació específica de negoci a dur a terme.
3. Invocar la operació de negoci o la delega a un manejador.
4. Ajuda a seleccionar la següent vista a mostrar al client.
5. Retorna la vista al client.

Aquesta separació simplifica enormement el disseny tant de vistes com de components del model. Les pàgines JSP no tenen que incloure maneig d'errors, mentre que els elements de control simplement decideixen sobre el pas següent.

Entre les característiques de Struts es pot mencionar:

- Configuració del control centralitzada.
- Les interrelacions entre accions i pàgines o altres accions s'especifiquen per taules XML enlloc de codificar-les en programes o pàgines.
- Components de l'aplicació, que són el mecanisme per compartir informació bidireccional entre l'usuari de l'aplicació i les accions del model.
- Llibreries d'entitats per facilitar la majoria de les operacions que generalment realitzarien les pàgines JSP.
- Struts conté una eina per la validació de camps de plantilles sota varis esquemes que van des de validacions locals en la pàgina (en JavaScript) fins les validacions de fons fetes a nivell de les accions.

Struts permet que el desenvolupador es concentri en el disseny d'aplicacions complexes com una sèrie simple de components del Model i de la Vista intercomunicades per un control centralitzat. Dissenyant d'aquesta manera es pot obtenir una aplicació més consistent i més fàcil de mantenir.

Per més informació consultar [2] i [4].

Annex C

JGraphT

JGraphT és una llibreria lliure de grafs de Java que proporciona d'objectes i algorismes matemàtics de la teoria de grafs. JGraphT soporta diferents tipus de grafs incloent:

- Grafs directes i indirectes.
- Grafs amb pesos / etiquetes o qualsevol aresta definit per l'usuari.
- Vaires opcions de multiplicitat de les arestes, incloent grafs simples, multigrafs, pseudografs.
- Grafs no modificables, permetent que els mòduls proporcionin el accés “nomes-lectura” als grafs interns.
- Grafs que poden ser escoltats, permetent que els oients externs segueixin els esdeveniments de les modificacions.
- Grafs subgrafs, que s'actualitzen sols d'altres grafs.
- Totes les composicions de grafs ja dits.

Per més informació consultar [11] i [12].

Explicació de les funcions utilitzades per desenvolupar el digraf de les preguntes:

Interface DirectedGraph<V,E>

- `E addEdge (V sourceVertex, V targetVertex)` : Crea una nova aresta en el graf, des de el `sourceVertex` fins al `targetVertex`.
- `boolean addVertex (V v)` : Afegeix el vertex especificat al graf si encara no està afegit.
- `boolean containsEdge (V sourceVertex, V targetVertex)` : Retorna cert si hi ha una aresta que va des de el `sourceVertex` fins al `targetVertex`.
- `V getEdgeTarge (E e)` : Retorna el vèrtex `targetVertex` de l'aresta.
- `Java.util.Set<E> incomingEdgesOf (V vertex)` : Retorna una col·lecció de totes les arestes que venen cap al vèrtex especificat.
- `Java.util.Set<E> ioutgointEdgesOf (V vertex)` : Retorna una col·lecció de totes les arestes que marixen del vèrtex especificat.
- `E removeEdge (V sourceVertex, V targetVertex)` : Elimina l'aresta que va des de el `sourceVertex` fins al `targetVertex`, si existeix l'aresta i els vèrtexs. Retorna l'aresta si s'elimina o null altrament.
- `boolean removeVertex (V v)` : Elimina el vèrtex `v` del graf. Retorna cert si el graf contenia el vèrtex.

Annex D

Especificació

Els mètodes de les classes Answer.java, Question.java, Resource.java, Result.java i Test.java són únicament mètodes getters i setters d'accés als atributs de les classes.

ServiceAnswer.java

- `public ServiceAnswer()`

Aquesta funció és la constructora de la classe.

- `public void addAnswer(Answer a)`

Aquesta funció serveix per afegir una resposta a la base de dades.

- `public Answer findAnswerById (int id)`

Aquesta funció serveix per obtenir una resposta sabent el seu identificador.

- `public List<Answer> findAnswersByQuestion(int idQuestion)`

Aquesta funció serveix per obtenir una llista de respostes associades al identificador d'una pregunta.

- `public void updateAnswer(Answer a)`

Aquesta funció serveix per actualitzar una resposta de la base de dades.

- `public void deleteAnswer(int id)`

Aquesta funció serveix per eliminar una resposta sabent el seu identificador.

- `public void deleteAnswersByQuestionId (int id)`

Aquesta funció serveix per eliminar les preguntes associades al identificador d'una pregunta.

ServiceDependency.java

- `public ServiceDependency()`

Aquesta funció és la constructora de la classe.

- `public void addDependency(int a, int b)`

Aquesta funció serveix per afegir una dependència entre la pregunta amb identificador *a* i la pregunta amb identificador *b*.

- `public List<Integer> getQuestionA(int id)`

Aquesta funció serveix per obtenir una llista de les preguntes que depèn la pregunta amb identificador *id*.

- `public List<Integer> getQuestionB(int id)`

Aquesta funció serveix per obtenir una llista de les preguntes que depenen de la pregunta amb identificador *id*.

- `public List<Integer> getPossiblesQuestionsA(int id,int o)`

Aquesta funció serveix per obtenir una llista de les possibles preguntes que poden dependre de la pregunta amb identificador *id*.

- `public void deleteQuestionA(int id)`

Aquesta funció serveix per eliminar totes les dependències de les preguntes que depèn la pregunta amb identificador *id*.

- `public void deleteQuestionB(int id)`

Aquesta funció serveix per eliminar totes les dependències de les preguntes que depenen de la pregunta amb identificador *id*.

- `public void delete(int A, int B)`

Aquesta funció serveix per eliminar la dependència entre la pregunta amb identificador *A* i la pregunta amb identificador *B*.

ServiceQuestion.java

- `public ServiceQuestion()`

Aquesta funció és la constructora de la classe.

- `public int addQuestion(Question q)`

Aquesta funció serveix per afegir una pregunta a la base de dades i obtenir el seu identificador.

- `public Question findQuestionById(int id)`

Aquesta funció serveix per obtenir una pregunta sabent el seu identificador.

- `public List<Question> findQuestionsByObjective(int idObjective)`

Aquesta funció serveix per obtenir una llista de les preguntes que pertanyen a un objectiu determinat.

- `public void updateQuestion(Question q)`

Aquesta funció serveix per actualitzar una pregunta de la base de dades.

- `public void deleteQuestion(int id)`

Aquesta funció serveix per eliminar una pregunta sabent el seu identificador. Això també implica eliminar les seves dependències i les seves respostes.

- `public List<Question> findQuestionsByTest(int idTest)`

Aquesta funció serveix per obtenir una llista de les preguntes que pertanyen a un test determinat.

- `public Question findQuestionByPositionInTest(int idTest, int pos)`

Aquesta funció serveix per obtenir la pregunta d'una posició determinada que pertany a un test determinat.

ServiceResource.java

- `public ServiceResource()`

Aquesta funció és la constructora de la classe.

- `public void addResource(Resource r)`

Aquesta funció serveix per afegir un recurs a la base de dades.

- `public Resource findResourceById (int id)`

Aquesta funció serveix per obtenir un recurs sabent el seu identificador.

- `public List findResourceByObjectiveAndType(int objective,String type)`

Aquesta funció serveix per obtenir una llista dels recursos d'un objectiu i un tipus determinat.

- `public void updateResource(Resource r)`

Aquesta funció serveix per actualitzar un recurs de la base de dades.

- `public void deleteResource(int id)`

Aquesta funció serveix per eliminar un recurs sabent el seu identificador.

ServiceResult.java

- `public ServiceResult()`

Aquesta funció és la constructora de la classe.

- `public void addResult(Result r)`

Aquesta funció serveix per afegir un resultat a la base de dades.

- `public Result findResultByTestIdAndQuestionId (int idTest, int idQuestion)`

Aquesta funció serveix per obtenir un resultat sabent el identificador del test i el identificador de la pregunta.

- `public void updateResult(Result r)`

Aquesta funció serveix per actualitzar un resultat de la base de dades.

ServiceTest.java

- `public ServiceTest()`

Aquesta funció és la constructora de la classe.

- `public int addTest(Test q)`

Aquesta funció serveix per afegir un test a la base de dades i obtenir el seu identificador.

- `public void addQuestionTest(int idTest, Question q)`

Aquesta funció serveix per afegir la pregunta *q* al test determinat.

- `public Test findTestById (int id)`

Aquesta funció serveix per obtenir un test sabent el seu identificador.

- `public List<Test> findTestByStudent(int idStudent)`

Aquesta funció serveix per obtenir una llista dels tests realitzats per un alumne determinat.

- `public ArrayList<Test> findTestByObjective(int idObjective)`

Aquesta funció serveix per obtenir una llista dels tests realitzats d'un objectiu determinat.

- `public void updateTest(Test t)`

Aquesta funció serveix per actualitzar un test de la base de dades.

- `public void deleteTest(int id)`

Aquesta funció serveix per eliminar un test determinat.

TestGenerator.java

- `public TestGenerator(int objective)`

Aquesta funció és la constructora de la classe. Se li passa com a paràmetre el identificador de l'objectiu del qual es vol generar el test.

- `public void generateTest(int idTest)`

Aquesta funció serveix per generar les preguntes d'un test.

Annex E

Preguntes des de fitxer

Un aspecte important sobre treball futur relacionat amb el meu mòdul i l'aplicació InnovaCampus és la possibilitat de donar d'alta preguntes des d'un fitxer. En aquest annex explicarem la manera de com fer-ho.

La idea consisteix en que tenir la possibilitat de donar d'alta totes les preguntes des de un fitxer determinat que triaries un cop estiguessis dintre l'aplicació.

Quan donem d'alta una pregunta, es necessita la següent informació:

- Nom de l'assignatura.
- Nom del tema.
- Nom de l'objectiu.
- Text pregunta.
- Respostes: text, comentari i saber si es correcta o no.
- Saber si te recurs associat i quin es.
- Saber si depèn d'alga altra pregunta.

El camp assignatura, tema i objectiu són camps optatius, ja que depèn de des de on tries el fitxer aquest camps ja els coneixes o no.

El camp text de la pregunta és obligatori, de la mateixa manera que el text de les respostes, els comentaris i saber si les respostes son correctes o no.

Els camps de lligar un recurs i les dependències són camps optatius, perquè una pregunta pot estar o no associada a un recurs i una pregunta pot dependre o no d'altres preguntes.

Un cop sabem la informació que heura de contenir el fitxer, vegem com s'ha d'estructurar aquesta informació. La manera més senzilla i fàcil és utilitzar etiquetes o marcadors per saber on comença cada camp d'informació. Es poden fer servir les següents etiquetes:

| | |
|-------------------------------|---------------------------|
| Nom assignatura | <Subject:name> |
| Nom tema | <Topic:name> |
| Nom objectiu | <Objective:name> |
| Text pregunta | <Question:text> |
| Text resposta | <Answer:text> |
| Comentari resposta | <Answer:commentary> |
| Resposta correcta | <Answer:correctTrue> |
| Resposta errònia | <Answer:correctFalse> |
| Tipo recurs | <Resource:type> |
| Nom recurs | <Resource:name> |
| Text pregunta dependent | <Dependency:questionText> |

Els passos ha seguir per utilitzar les etiquetes en un fitxer són:

1. Si cal, escriure l'etiqueta <Subject:name> i el nom de l'assignatura.
2. Si cal, escriure l'etiqueta <Topic:name> i el nom del tema.
3. Si cal, escriure l'etiqueta <Objective:name> i el nom de l'objectiu.
4. Escriure l'etiqueta <Question:text> i el text de la pregunta.
5. Escriure l'etiqueta <Answer:text> i el text de la resposta.
6. Escriure l'etiqueta <Answer:commentary> i el comentari de la resposta.
7. Si es vol que aquesta sigui una resposta correcta s'haurà d'escriure <Answer:correctTrue>, però si es vol que sigui una resposta errònia es pot escriure l'etiqueta <Answer:correctFalse> o no escriure res.
8. Si es volen incloure més respostes tornem al pas 5, si ja estan totes les respostes continuem.
9. Si es vol associar un recurs, escriuríem l'etiqueta <Resource:type> i el tipus de recurs que es, i l'etiqueta <Resource:name> i el nom del recurs.
10. Si es vol associar una dependència a una altra pregunta escriuríem l'etiqueta <Dependency:questionText> i el text de la pregunta corresponent. Se sobreentendrà que l'assignatura, tema i objectiu d'aquesta assignatura són els mateixos que els de l'assignatura que estem donant d'alta.
11. Un cop s'han completat la pregunta, per procedir a la següent pregunta dins el fitxer, tornaríem al pas 1.

Aspectes a considerar:

- Si al començament del fitxer ja has ficat l'assignatura, tema i/o objectiu, no caldrà repetir-ho en cada pregunta, s'entendrà que es l'últim valor introduït.
- Si no hi ha cap resposta o totes les respostes són errònies la pregunta no es donarà d'alta i s'avisarà a l'usuari.
- En cas que al recurs li falti algun dels dos camps (tipus o nom) o que el recurs no existeixi, no s'associarà cap recurs a la pregunta i s'avisarà a l'usuari.
- En cas de voler associar una dependència i escriure el text d'una resposta inexistent pel objectiu establert, la dependència no s'efectuarà i s'avisarà a l'usuari.
- En cas de produir-se un error dels no nombrats anteriorment, no es donaran d'alta les preguntes i s'avisarà a l'usuari.
- Es podria fer que si se vol lligar un recurs o una dependència, que s'escrigués una etiqueta especial i en acabar de donar d'alta totes les preguntes, s'obrís en mode edició totes les preguntes que es vol lligar un recurs o una dependència.
- Es podria fer que l'usuari pugues triar el nom de les etiquetes, per així donar més dinamisme i intentar adaptar-se més als diferents dipus de fitxers.

Annex F

Llicència GPL

Utilitzarem la llicència GPL per al nostre mòdul del projecte InnovaCampus.

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The

resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a

particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work,

and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This

License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your

reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be

resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will

automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling

its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the

patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local

law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Llistat d'acrònims

| | |
|-------|--|
| ASP | Active Server Pages |
| CSS | Cascading Style Sheets |
| GPL | General Public License |
| HTML | Hyper Text Markup Language |
| HTTP | Hypertext Transfer Protocol |
| ISO | International Organization for Standardization |
| JRE | Java Runtime Enviroment |
| JSP | JavaServer Pages |
| LGPL | Lesser General Public License |
| PHP | Hypertext Preprocessor |
| SVG | Scalable Vector Graphics |
| URL | Uniform Resource Locator |
| UTF | Unicode Transformation Format |
| XHTML | eXtensible Hypertext Markup Language |
| XML | eXtensible Markup Language |
| W3C | World Wide Web Consortium |

Bibliografia

- [1] Alet Alís, Marcel: *InnovaCampus: Una aplicació Web d'ajut a l'avaluació en una titulació universitària*.
Treball de final de carrera. ETIG, EPS, UDL 2006-2007

- [2] Apache Struts: *Struts*.
<http://struts.apache.org/>
Darrer accés: 12 de juliol del 2007

- [3] Canales, Ivan: *Apunts sobre JavaScript del bloc d'Internet*.
UDL 2006-2007

- [4] Cavaness, Chuck: *Jakarta Struts*.
O'REILLY. 2005

- [5] Ciències de la Computació i Intel·ligència Artificial: *Curs d'HTML de la Universitat d'Estiu (UdL)*
<http://fermat.eup.udl.es/UdL-Estiu/>
Darrer accés: 12 de juliol del 2007

- [6] Java Sun: *Java Technology*.
<http://java.sun.com/j2se/1.5.0/docs/api/>
Darrer accés: 12 de juliol del 2007

- [7] Java Sun: *Api Java 2 Platform Standard Edition 5.0*.
<http://java.sun.com>
Darrer accés: 12 de juliol del 2007

- [8] Java Sun: *Learning the Java Language (The Java Tutorilas)*.
<http://java.sun.com/docs/books/tutorial/java/index.html>
Darrer accés: 12 de juliol del 2007

- [9] Jeroen Wijering: *JW FLV Player*.
http://www.jeroenwijering.com/?item=JW_FLV_Player
Darrer accés: 15 de juny del 2007

- [10] Jeroen Wijering: *JW MP3 Player*.
http://www.jeroenwijering.com/?item=JW_MP3_Player
Darrer accés: 13 de juny del 2007

- [11] JGraphT – a free Java Graph Library: *Welcome to JGraphT*.
<http://jgrapht.sourceforge.net/>
Darrer accés: 18 de juliol del 2007

- [12] JGraphT – a free Java Graph Library: *Javadocs*.
<http://www.jgrapht.org/javadoc/>
Darrer accés: 18 de juliol del 2007

- [13] FFmpeg
<http://ffmpeg.mplayerhq.hu/>
Darrer accés: 14 de juny del 2007

- [14] Fundación Sidar - Accesibilidad Universal: Hojas de Estilo en Cascada
<http://www.sidar.org/recur/desdi/mcss/manual/indice.php>
Darrer accés: 12 de juliol del 2007

- [15] Galbraith B., Falknet J., Irani R.: *Desarrollo Web con JSP*.
Ed. Anaya Multimedia. 2002

- [16] Wiki – Viquipèdia: *Portada – Viquipèdia*.
<http://ca.wikipedia.org/wiki/Portada>
Darrer accés: 14 de juny del 2007

- [17] W3Schools: *Online Web Tutorials*.
<http://www.w3schools.com/>
Darrer accés: 14 de juny del 2007

